Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations 1. Thesis and Dissertation Collection, all items

1977-12

Design and construction of a flight monitor and data recorder.

Kane, Dennis Leland

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/18068

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

DESIGN AND CONSTRUCTION OF A FLIGHT MONITOR AND DATA RECORDER

Dennis Leland Kane

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

DESIGN AND CONSTRUCTION OF A FLIGHT
MONITOR AND DATA RECORDER

by

Dennis Leland Kane

December 1977

Thesis Advisor: Uno R. Kodres

Approved for public release; distribution unlimited.

T182113

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Design and Construction of a Flight Monitor and Data Recorder

5. TYPE OF REPORT & PERIOD COVERED
Master's Thesis;
December 1977

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Dennis Leland Kane

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93940

12. REPORT DATE
December 1977

13. NUMBER OF PAGES
95

14. MONITORING AGENCY NAME & ADDRESS (If different from Controlling Office)
Naval Postgraduate School
Monterey, California 93940

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Flight Monitor
Single Board Computer
Magnetic Bubble Memory

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The design and preliminary testing of a microcomputer based flight monitor and data recorder, utilizing magnetic bubble memory, is reported. Component selection, software design and magnetic bubble storage system construction and testing are discussed. Difficulties encountered, both in software and bubble testing are reviewed, with results and remaining work summarized.

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601

Magnetic bubble memory technology is reviewed and its potential as a reliable, dense, low cost, non-volatile recording medium is noted. It is proposed that the microprocessor be utilized as a flight monitoring as well as a recording device to detect and report imminent "extremis" situations. This research is continuing at the Naval Postgraduate School.

Approved for public release; distribution unlimited

Design and Construction of a Flight Monitor and Data Recorder

by

Dennis Leland Kane
Lieutenant, United States Navy
B.S.E.E., University of New Mexico, 1971

Submitted in partial fulfillment of the
requirements for the degree of

Master of Science in Aeronautical Engineering

from the
NAVAL POSTGRADUATE SCHOOL
December 1977

Thesis
K136
c.1

ACKNOWLEDGMENTS

I would like to thank Texas Instruments Corporation for the technical assistance they so willingly supplied and for the use of diagrams and transparencies on magnetic bubble memories. I would like to thank Intel Corporation for its patient assistance in getting my SBC 80/20 wired up. I express my appreciation to the personnel of the Air Force Avionics Laboratory for the information supplied both in references and in informal exchange. I wish also to thank my thesis advisor, Dr. Uno Kodres, and Captain Clyde Tuomela, USN, for their patience, guidance and support which made this report possible. And finally, I would like to express my deep appreciation to my wife, Kathy, without whose support and encouragement over the past two years this report and my other work could not have been achieved.

ABSTRACT

The design and preliminary testing of a microcomputer based flight monitor and data recorder, utilizing magnetic bubble memory, is reported. Component selection, software design and magnetic bubble storage system construction and testing are discussed. Difficulties encountered, both in software and bubble testing are reviewed, with results and remaining work summarized.

Magnetic bubble memory technology is reviewed and its potential as a reliable, dense, low cost, non-volatile recording medium is noted. It is proposed that the microprocessor be utilized as a flight monitoring as well as a recording device to detect and report imminent "extremis" situations. This research is continuing at the Naval Postgraduate School.

LIST OF ACRONYMS

ALTC	Altitude-Course
ALTF	Altitude-Fine
BCLK/	Bus Clock
BDEN	Board Enable
BDSEL	Board Select
BIT	Binary Digit
BYTE	Eight Bits
CPU	Central Processing Unit
DATAEN	Data Enable
DBIN	Data Bus In
DIP	Dual In-Line Package
ENDCK	End Check
EPROM	Electrically Programmable Read Only Memory
FIFO	First In-First Out
GDATA	Good Data
ICE	In Circuit Emulator
I/O	Input-Output
ISIS	Intel System Implementation Supervisor
K	1024
KHZ	Kilohertz
LSI	Large Scale Integrated Circuit
M	Mega or $(1024)^2$
MBM	Magnetic Bubble Memory
MDS	Microcomputer Development System

MEMEN	Memory Enable
MIL STD	Military Standard
MNOS	Metal Nitride Oxide Semiconductor
MRDC	Memory Read Command
MWRC	Memory Write Command
NOP	No-Operation
PFWP	Power Fail Warning
PLM	Programming Language for Microcomputers
RAM	Random Access Memory
RDCTRCL	Read Counter Clear
ROM	Read Only Memory
SBC	Single Board Computer
STACKPTR	Stack Pointer
TFG	Timing Function Generator
TI	Texas Instruments
TTL	Transistor-Transistor Logic
VDC	Volts Direct Current

TABLE OF CONTENTS

I. INTRODUCTION -- 12

II. SYSTEM DESCRIPTION --------------------------------------- 17

- A. GENERAL -- 17
- B. MICROCOMPUTER SYSTEM --------------------------------------- 17
- C. MAGNETIC BUBBLE MODULE ------------------------------------ 24
  - 1. Magnetic Bubble Memory Board ---------------------------- 24
  - 2. Controller Board -- 29

III. CAPABILITY EXPANSION ------------------------------------- 39

- A. INTERRUPT VERSUS STATUS CHECK ------------------------------ 39
- B. SAFETY PROGRAMMING --- 40

IV. RESULTS AND RECOMMENDATIONS ------------------------------- 43

APPENDIX A BUBBLE TECHNOLOGY ---------------------------------- 45

- A. INTRODUCTION --- 45
- B. CONSTRUCTION --- 48
- C. APPLICATIONS --- 49

APPENDIX B COMPUTER PROGRAMS ---------------------------------- 54

- A. GENERAL -- 54
- B. RECORD PROGRAM --- 57
  - 1. Real Time Analysis -------------------------------------- 57
  - 2. Parameter Analysis and Compression ---------------------- 58
  - 3. MBM Call -- 58
- C. BUBBLE DRIVER PROGRAM -------------------------------------- 58
- D. INTERRUPT CONTROLLED BUBBLE TEST PROGRAM ------------------- 59

APPENDIX C CONSTRUCTION NOTES --------------------------------- 83

- A. GENERAL -- 83

B. MICROCOMPUTER ---------- 83

  1. Fail Safe Timer ---------- 83

  2. EIA Interface ---------- 84

  3. Interrupt Implementation ---------- 84

  4. Master Bus ---------- 84

  5. System Reset ---------- 84

C. CONTROLLER BOARD ---------- 84

  1. Interrupt ---------- 85

  2. Counters ---------- 85

  3. ROMs ---------- 85

  4. Addressing ---------- 85

  5. Wiring ---------- 85

  6. Power Failure ---------- 85

D. MBM BOARD ---------- 85

  1. Capacitors ---------- 86

  2. Diodes ---------- 86

E. INPUT/OUTPUT ---------- 86

F. EXTERNAL SUPPLY ---------- 87

BIBLIOGRAPHY ---------- 94

INITIAL DISTRIBUTION LIST ---------- 95

LIST OF TABLES

TABLE I Redundancy Map ---------------------------------- 35

TABLE II Evaluation Card Function Timing PROM Contents ('S471) ---------------------------------- 38

TABLE III TMS 9916/TBM 0101 Parameters - -------------------- 53

TABLE IV Discrete Parameters ------------------------------ 77

TABLE V Data Recorder Memory Map -------------------------- 80

TABLE VI MDS Memory Map Under ICE/80 ---------------------- 82

TABLE VII Address Line Correspondence ---------------------- 88

TABLE VIII Interface Board Schematic ------------------------ 89

TABLE IX Controller Cage Backplane ------------------------ 91

TABLE X Cable-Interface Board to Controller --------------- 92

TABLE OF DRAWINGS

FIGURE 1 Microcomputer Safety Monitor and Flight Data Recorder -- 15

FIGURE 2 Safety Monitor/Data Recorder System ---------------- 16

FIGURE 3 SBC 80/20-4 -- 18

FIGURE 4 Interface Board ------------------------------------ 19

FIGURE 5 ICOM PROM Programmer ------------------------------- 20

FIGURE 6 Block Diagram of Microcomputer Group --------------- 21

FIGURE 7 Magnetic Bubble Module ----------------------------- 23

FIGURE 8 MBM Prototype Board -------------------------------- 26

FIGURE 9 Magnetic Bubble Memory Evaluation Board ------------ 27

FIGURE 10 MBM Controller Board ------------------------------ 30

FIGURE 11 MBM Controller Block Diagram ---------------------- 31

FIGURE 12 Magnetic Bubbles and Tracks ----------------------- 51

FIGURE 13 TBM 0101 -- 52

FIGURE 14 Development System -------------------------------- 55

FIGURE 15 Initialization ------------------------------------ 78

FIGURE 16 Single Page Read or Write ------------------------- 79

I. INTRODUCTION

A. BACKGROUND

The need for a light, crash survivable, compact data recorder for Naval aircraft is well established. Information which identifies the cause of the loss of one aircraft can often point to a weakness that may affect several others. Recognizing this formally, the Chief of Naval Operations has required the inclusion of crash recorders in all new Navy aircraft.

A crash recorder must utilize a recording medium that is non-volatile, as the data may not be recovered for weeks or months after being recorded. Additionally, if the recording medium is reusable, then no service action is required to install the new recording medium.

In the past, obstacles to the solution of the recording problem have included the cost, the reliability of mechanical recorders, the size and weight of any system proposed for a small aircraft, and the mass memory size required when implemented by a solid state system.

Reference [1] demonstrated that by ignoring redundant data, or "compressing" the data, one can reduce the size of the required memory by 25%-50%.

Solid state non-volatile data storage using an MNOS (Metal Nitride Oxide Semiconductor) module was demonstrated in Reference [2]. Until recently solid state non-volatile memories have been too bulky. Solid state Magnetic Bubble Memory (MBM) overcomes these problems by offering a quantum step forward in size, weight, cost/bit and reliability. By combining the processing power of the microcomputer with the dense memory

capacity of the MBM, a low cost, small size, highly reliable recording system becomes available for multipurpose use.

Because the processing speed of the microcomputer exceeds what is needed for recording data, and because system status data is available, the system can also be used as a flight monitor to aid the pilot in making critical decisions.

During an aircraft launch, approach, or combat engagement, decision time is critical. Analysis of pertinent factors relating to take-off abort, wave off, or ejection may require split second pilot decisions. The speed and power of the microcomputer can be used to monitor the aircraft state and greatly assist the pilot in such situations. Utilizing the same equipment and data as the recorder, no additional cost other than software is incurred for a substantial increase in capability.

B. THE MONITOR/RECORDER SYSTEM

This thesis consists of the system design and construction of a development prototype microcomputer monitor and data recorder. A non-volatile MBM module is constructed to function as a remote mass memory through appropriate interface circuitry. The memory module is designed so that it can be located in a survivable airfoil, and thus the system can function as a crash recorder.

The present system utilizes current production equipment for all components. With the advent of single chip controllers implementing CPU, CLOCK, EPROM/RAM memory and I/O functions on a single chip, future size reductions will be significant. Future system design centered

around such devices utilizing a single 400 cycle power supply can incorporate all required computer functions, except input bus interface, within 2-3 chips. Replacement of the current controller board and MBM prototype board with 3-5 LSI chips as these modules come into greater use will similarly reduce the size of this part of the system. Implementation of the complete unit on a 5" x 7" circuit board, with out power supply and input bus interface, is a realistic expectation within one to three years.

Section II gives an overview of the constructed digital data recording system, as well as the peripheral equipment interfaced to it for field test and modification.

Section III discusses interrupt versus status checks for software control. Important additional capabilities of the system, with recommendations for future implementation are also discussed.

Appendix A discusses bubble memory technology, with notes relating to some of its critical parameters. The developed programs are listed in Appendix B. Appendix C contains detailed hardware information, schematics and peripheral systems interface information.

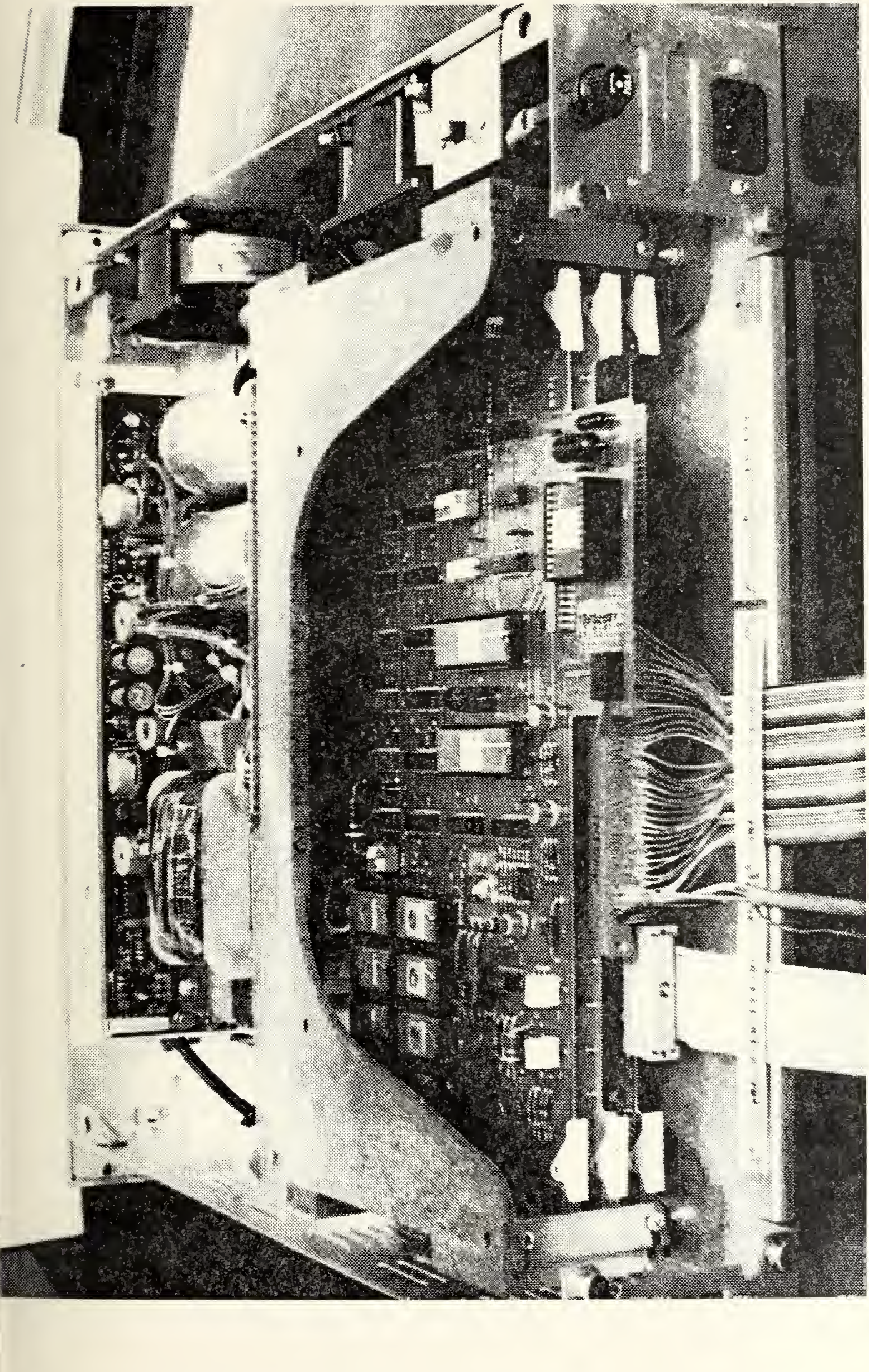


FIGURE 1

MICROCOMPUTER SAFETY MONITOR AND FLIGHT DATA RECORDER

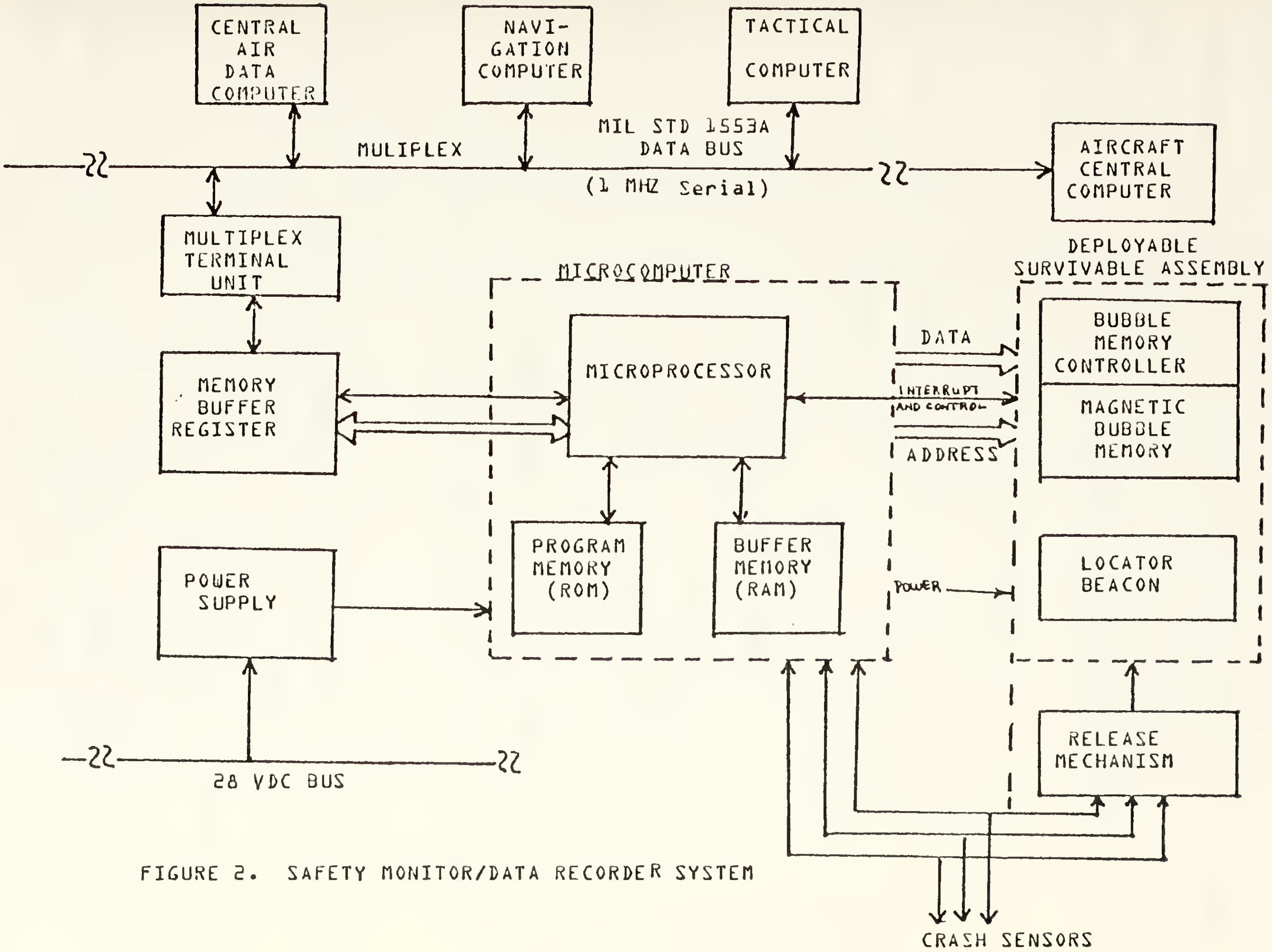


FIGURE 2. SAFETY MONITOR/DATA RECORDER SYSTEM

II. SYSTEM DESCRIPTION

A. GENERAL

The microcomputer monitor and data recording system is illustrated in Figure 1 and is comprised of two major assemblies; a general purpose microcomputer system and a remote magnetic bubble memory module. Figure 2 depicts the system at a block level.

As depicted in Figure 2, the system is designed to receive aircraft status data that is assumed to be received on a MIL STD 1553 data bus. This information is buffered and sent to the general purpose microcomputer. The microcomputer analyzes the data, compresses it, and sends the compacted data to storage in the magnetic bubble module.

The MBM controller receives the data from the computer. Here it is converted to serial data and stored in the bubble memory. The MBM module is part of an escape capsule that is designed to be survivable in the event of aircraft crash. (Reference [2] goes into greater detail in this area.) It is housed with a locater beacon to facilitate its recovery.

B. MICROCOMPUTER SYSTEM

The general purpose microcomputer system consists of an Intel 80/20-4 Single Board Computer (SBC) mounted in a rack suitable for a total of four boards. An integral DC supply provides all required power with the exception of +17 volts. This deficiency will be discussed in Appendix C. An ICOM model PP80 MDS/SBC 80/20-R PROM programmer board with resident software and 7K of expansion EPROM has been added in the card case to provide resident program modification capability, as well as additional

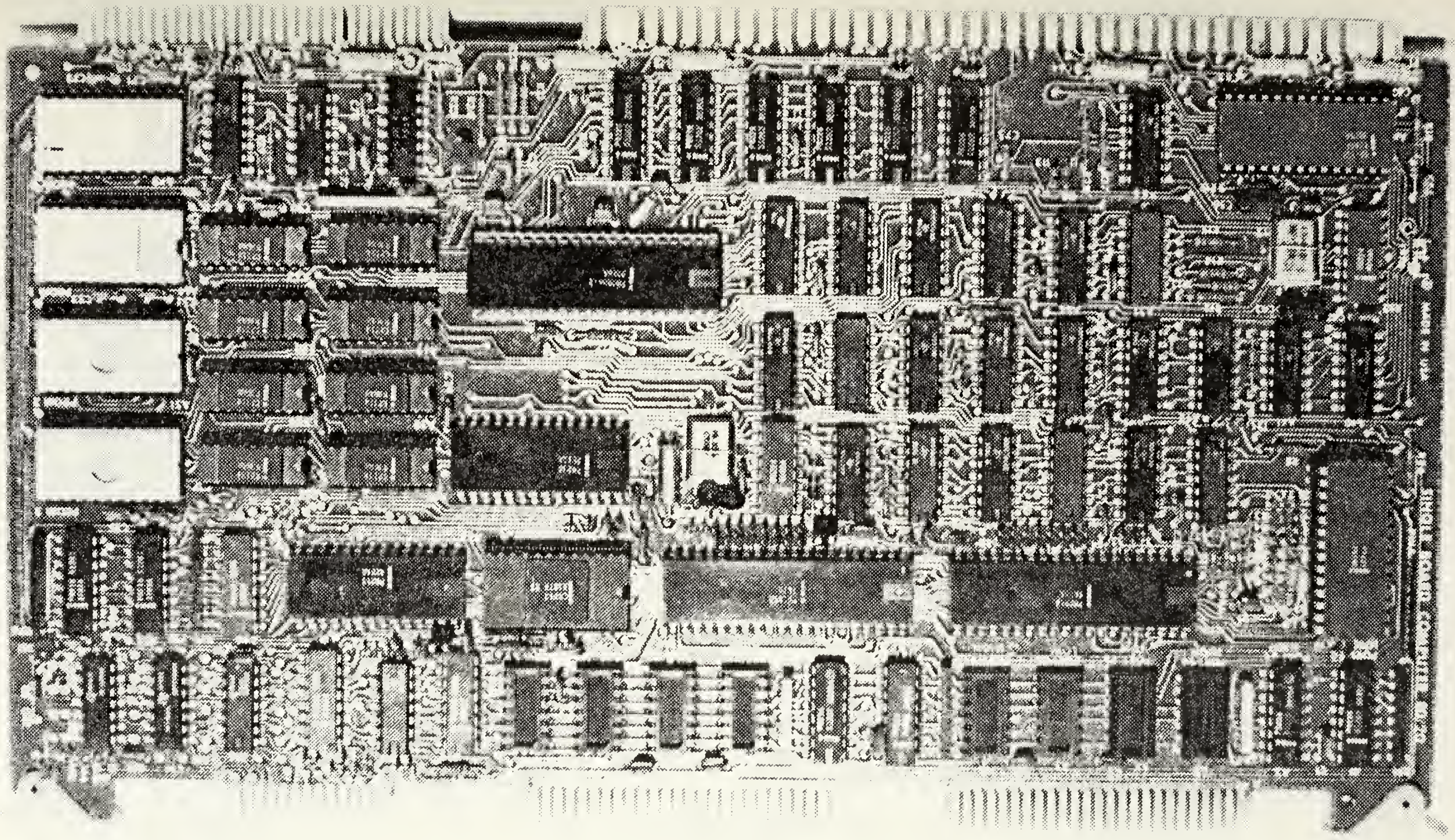


FIGURE 3

SBC 80/20-4

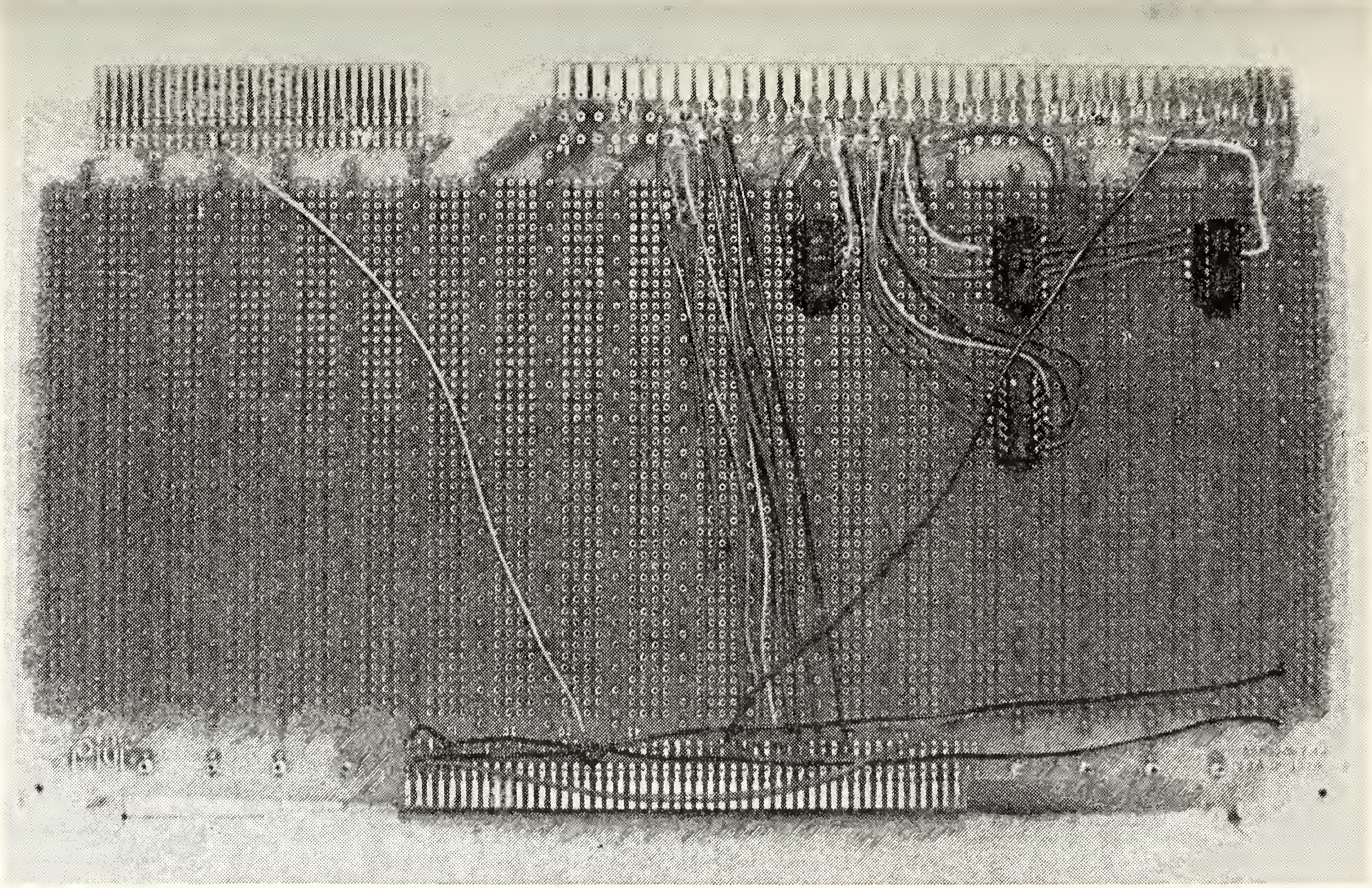


FIGURE 4

INTERFACE BOARD

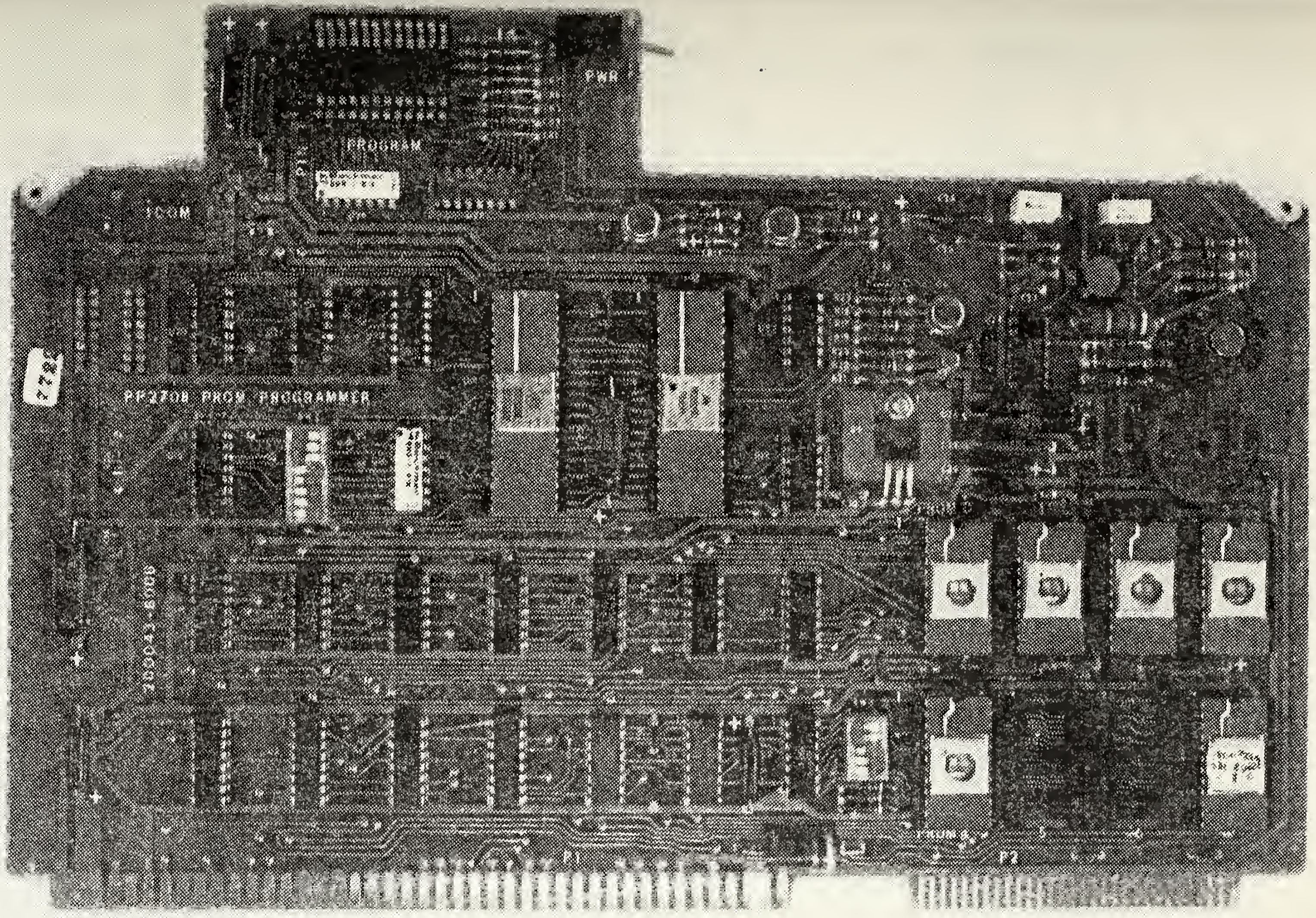
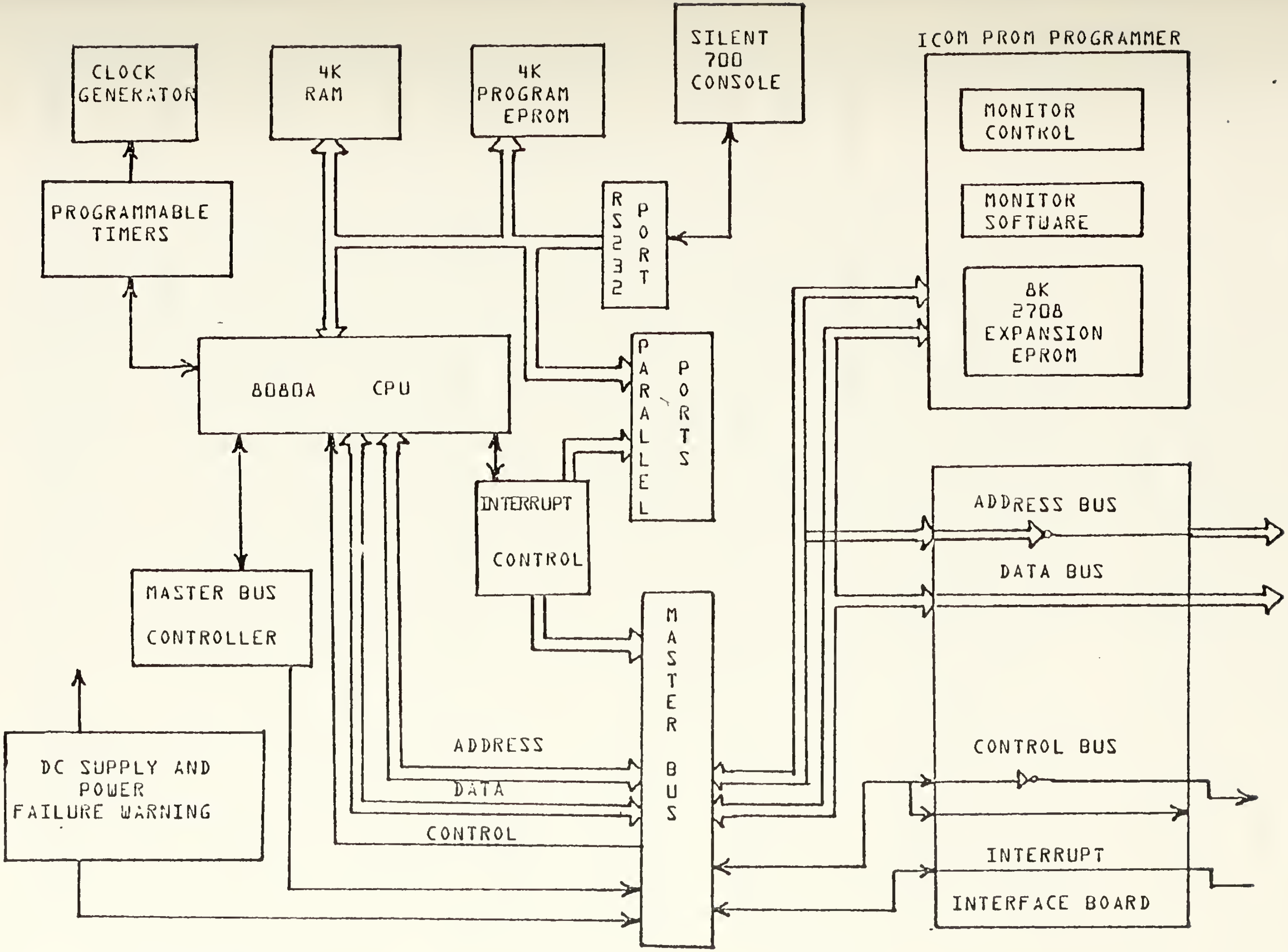


FIGURE 5

ICOM PROM PROGRAMMER



SYSTEM 80/20

FIGURE 6

BLOCK DIAGRAM OF MICROCOMPUTER GROUP

program storage. The third board in the cage is an interface board to accomplish all electrical interface to the remote magnetic bubble memory module. There is room for one more board in the card cage. It is anticipated that the MIL STD 1553 data bus interface would go on this in the future.

The Intel 80/20-4 Single Board Computer (SBC) is the heart of the system. The System 80/20-4 Microcomputer Hardware Reference Manual (Preliminary) [Reference (3)], and the SBC 80/20-4 Single Board Computer Hardware Reference Manual [Reference (10)], discuss in detail the many facets of its operation. Figures 3, 4, and 5 illustrate the boards within the computing system. Figure 6 depicts the system in a detailed block diagram, as configured for this project.

The System 80/20 was chosen due to its flexible interrupt structure, its power failure warning circuitry, and its multi-master bus configuration. With 4K bytes of resident EPROM, (2K of which is system monitor), 7-8K bytes of expansion EPROM on the ICOM board, as well as 4K bytes of resident RAM, there are no effective memory limitations imposed.

The system is set up to operate on its monitor utilizing automatic baud rate selection to an RS-232 serial interface. A Texas Instruments Silent 700 Terminal was obtained and modified to allow portable system operation. Terminal modification and wiring are indicated in Appendix C.

Field test and software modification is supported by the ICOM PROM programmer and Texas Instruments Silent 700 portable terminal. Program verification and alteration are immediately available to allow custom interface or software alteration, as well as bubble down-load under

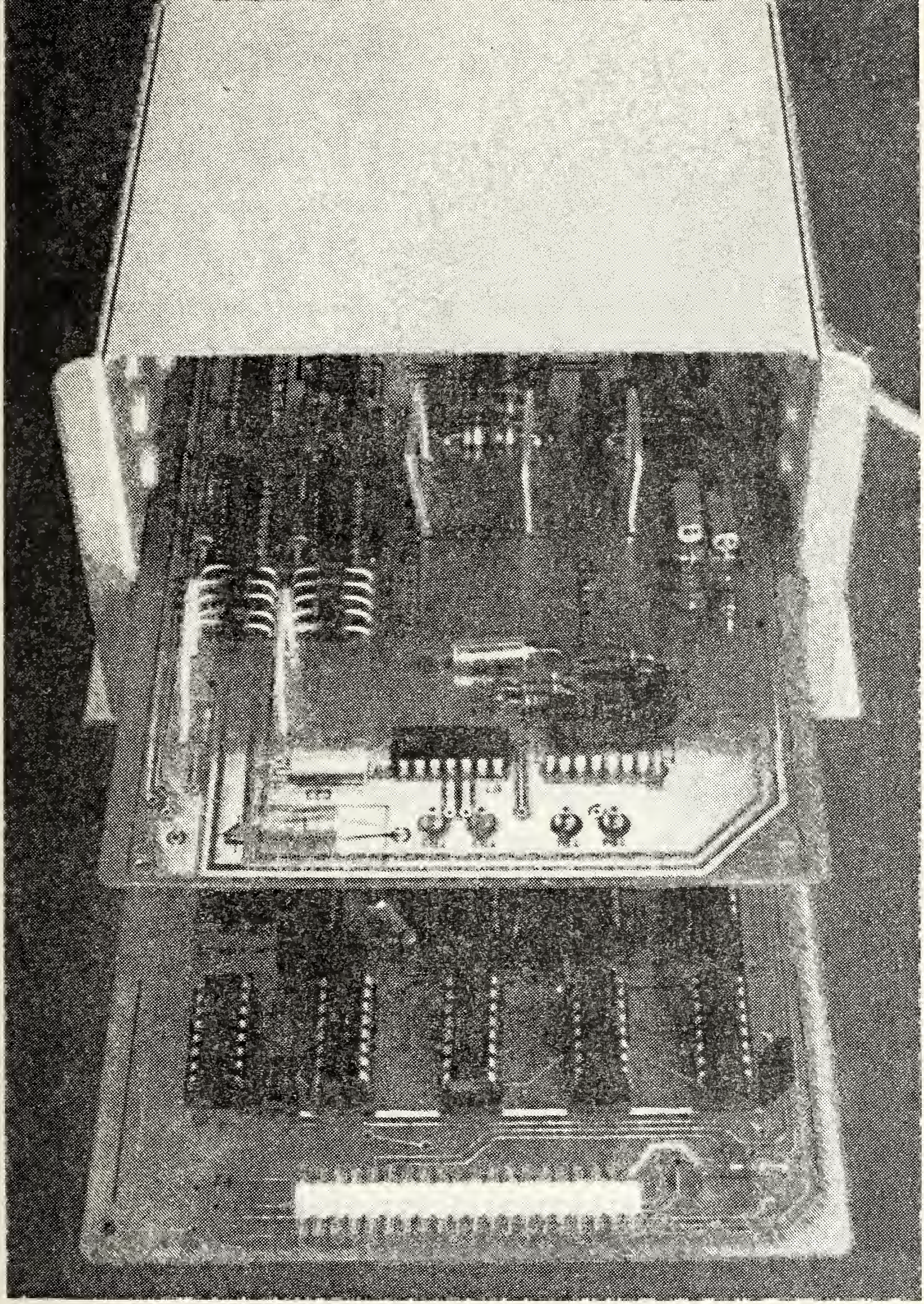


FIGURE 7

MAGNETIC BUBBLE MODULE

program control in the flight or post flight environment. ICOM PROM programmer operation is detailed in the PROM Programmer Operations Manual [Reference (4)]. Terminal operation is outlined in the Silent 700 Model 745 Terminal Operating Instructions [Reference (9)].

C. MAGNETIC BUBBLE MODULE

The magnetic bubble memory module is illustrated in Figure 7 and consists of two cards, housed together in a separate card cage. A forty-lead flat ribbon signal connector and a five-lead power cable connect to the interface board. The module consists of a magnetic bubble controller card and a magnetic bubble driver/sense amplifier card. Back-plane connections in the card cage transfer all required signals between the two boards.

Figures 9 and 11 are detailed block diagrams of the boards. The cards were locally built, utilizing first design printed circuit boards and schematic information from Texas Instruments (TI). These boards interface the TI TBM0101 magnetic bubble chip to the microcomputer as a TTL compatible interface. References [5] and [6] are the controller and bubble board detailed specification. References [7] and [8] are the electrical schematics and parts lists for each board.

1. Magnetic Bubble Memory Board

The Magnetic Bubble Memory (MBM) board is illustrated in Figure 8 and detailed at the block level in Figure 9. Detailed operation is outlined in Magnetic Bubble Memory and System Interface Circuits (Reference [6]). An overview of MBM board operation follows.

Board enable (BDEN) is input low, producing an enable to the coil field drives, transfer, replicate, generate, and annihilate gates, and to

the output data driver. U7 and U8 are indirectly controlled via U4 enable.

Field drive signals (CXA, CXB, CYA, CYB) are input to U1/BE from the controller board coil drive output circuitry. These precisely timed square wave pulses are applied to the appropriate coil drives and cause triangular current output pulses from the field drivers. These outputs are applied to the X and Y coil drives out of phase to produce the 100 KHZ field as described in Reference [6]. The timing of these field drive signals is controlled by the controller board. Only level conversion and shaping are done on the bubble board.

Monitoring of bubble loop position is accomplished by the controller. It assumes the bubble is started from a zero page reference and as such it must be allowed to return to this zero reference prior to shutdown or powerdown. The time involved to return the loops to the zero reference can vary from 0 to 6.4 ms depending on loop positions at commencement of shut down.

Chips U2A and U2B are the Transfer gate drivers. U2A and U2B are functionally identical and drive the same gates. Timing differences in the micro sequence from the controller determine whether the pulse in the transfer loops will perform a transfer in or a transfer out. Chip U3A contains the Replicate driver, while U3B contains the Annihilate driver. Again these two chips are schematically identical and drive a common gate. The difference in their function is accomplished via timing. Along with U4A, the combination of Replicate, Annihilate and Generate accomplish the bubble read/write function. The chips themselves are identical as are the output drive transistors. Control signals for the MBM board

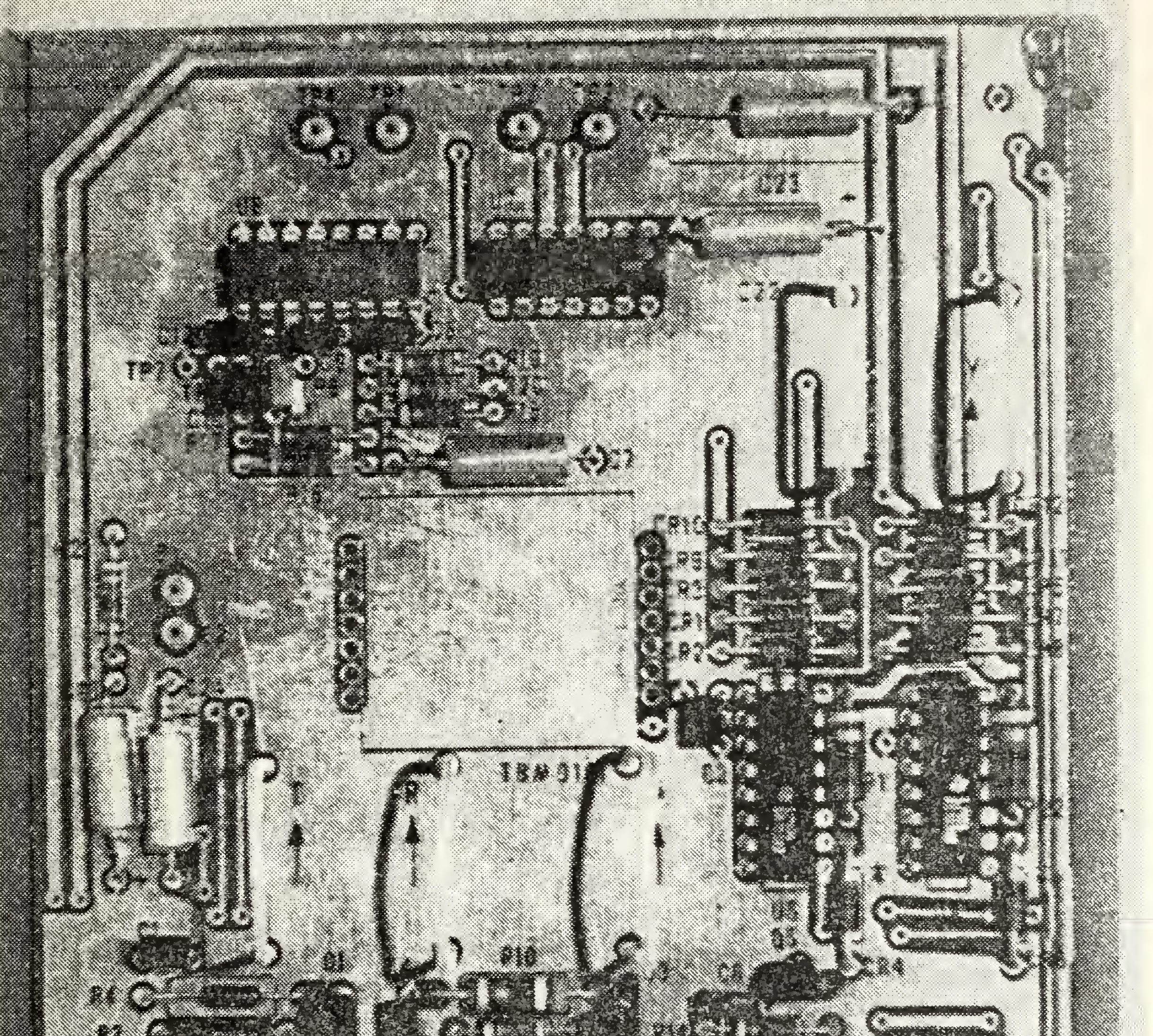
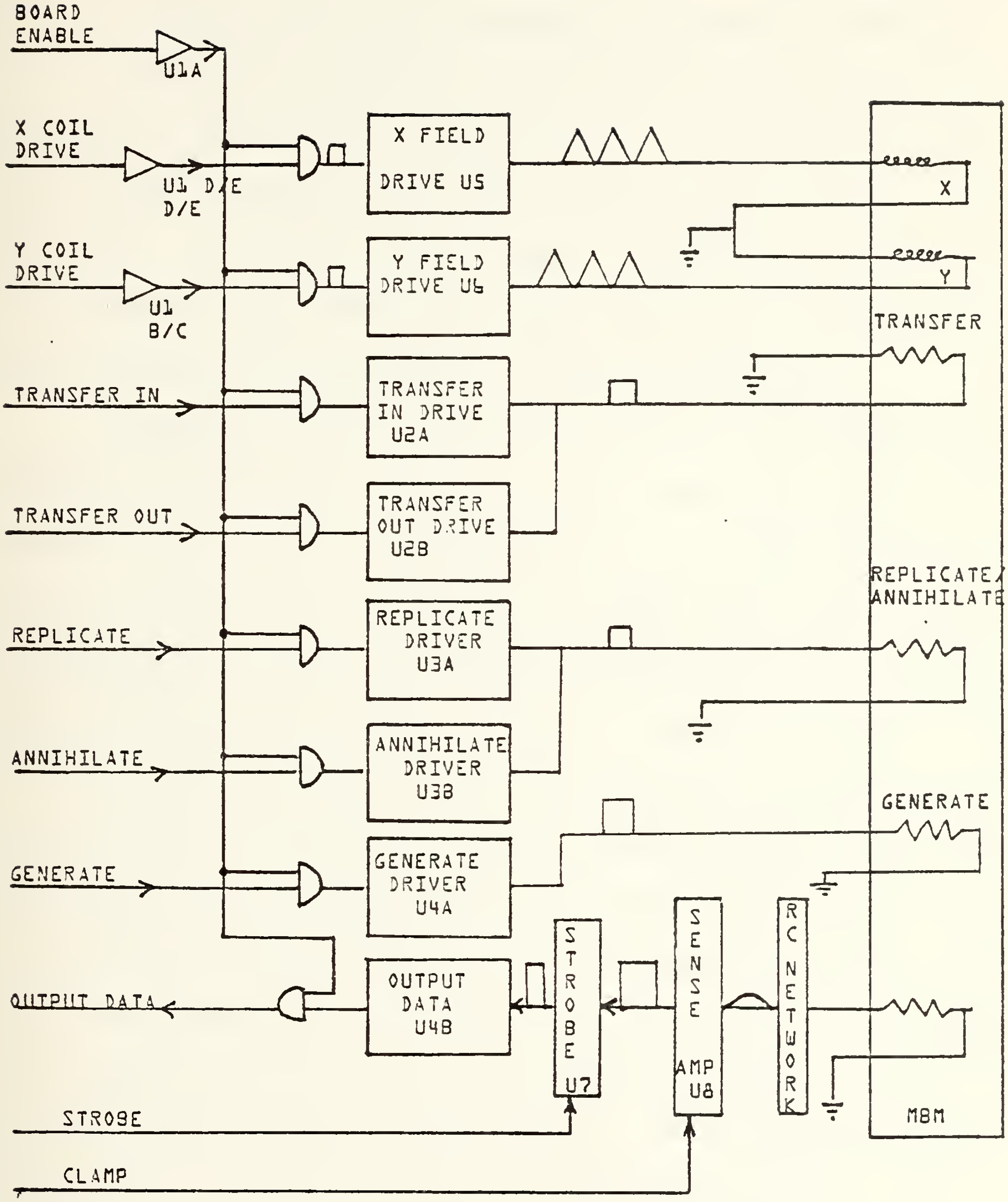


FIGURE 8

MBM PROTOTYPE BOARD



MAGNETIC BUBBLE MEMORY EVALUATATION BOARD FIGURE 9

are received as inputs from the Controller board. The Timing Function Generator sequences the chips to accomplish all required functions.

Output data sensing is accomplished via U4B, U7, U8 and the R/C network. When an analog bubble signal is sensed in the detectors it is coupled to the sense amplifiers, amplified and digitized via the clamp signal. The sense amplifier output is applied to latch U7 where it is strobed to the output driver, U4B. The output of this driver is then transferred to the controller as digital bubble data.

Reference [6] points out that the gates in the bubble memory chip make excellent fuses due to their small size. Discussion with TI personnel bears this out as a major failure area For this reason a resistive equivalent for the bubble chip elements was constructed for the testing phase, utilizing specification information of Reference [6]. Prior to bubble chip insertion, all signals should be verified for timing. In particular, the polarity, pulse width, and duty cycle of the gate drive waveforms should correspond with those of Reference [6]. The five large wire loops are for final verification of current waveforms prior to bubble chip insertion, and for verification with the bubble installed. Utilizing a current probe, the circuit waveforms may be verified at the bubble. At test completion these loops may be reduced to straight wire runs to the bubble chip, for a more compact design. All timing waveforms are referenced with respect to the leading (falling) edge of CXB/, Pin 18. This signal, input as an external sync, set for negative slope trigger, will be required for proper test equipment timing.

2. Controller Board

No written information on the controller board was available. Reference [6] suggests a design and it appears the board was modeled after this. The board is centered around a TMS 9916 Magnetic Bubble Memory Controller. Figure 9 illustrates the board itself, while Figure 11 is the block diagram. All three busses, data, address and control, are used as inputs. The control bus is made up of five signals: reset, clock (BCLK/), power fail (PFWP/), data bus in (DBIN), memory enable (MEMEN) and Interrupt. Hardware reset is a low level signal that is tapped directly from the SBC reset signal. It performs a full reset of the controller board via hardware. SBC BCLK/ was selected as the external clock input to the controller, since all data output to the master bus of the SBC is referenced to this clock. This TTL signal is input to U23 and U19 to properly sync the ready and read/write operations between the CPU and controller.

Power fail (PFWP/) is a low level input signal from the system 80/20. Reference [3] describes its operation. It is currently tied to +5V on the card cage backplane. To utilize it, PFWP/ would be brought from the system 80/20 to the power bad input of the controller board, P1/19, as a low logic signal. This circuit is designed in the controller to allow an immediate orderly shutdown of the bubble memory in the event of power failure, because major loop data loss is possible if data is left in it at shutdown. Additionally, to keep track of zero page reference, the bubble must shut down with the minor loops positioned at a known point with respect to the transfer gates.

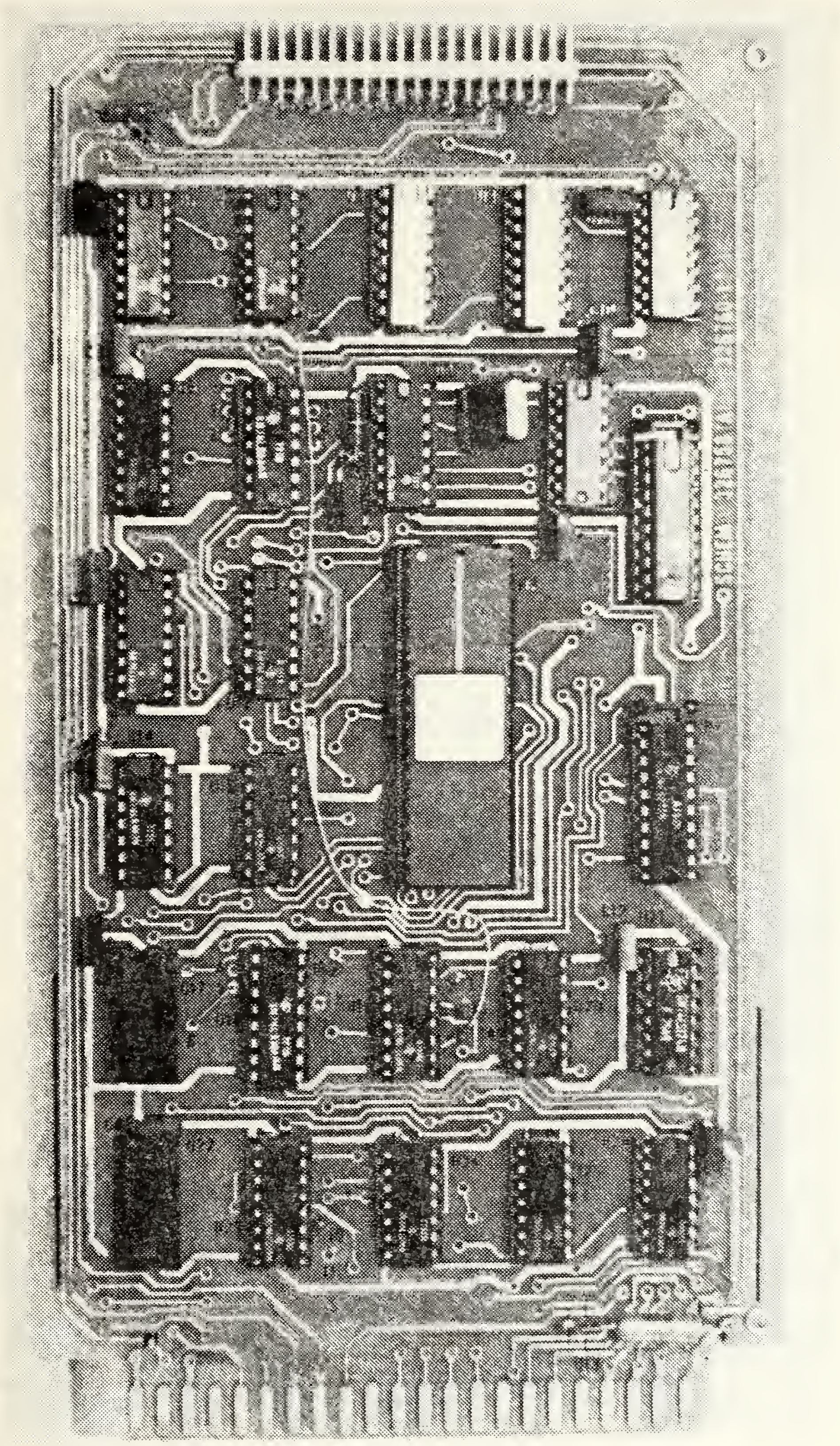


FIGURE 10

MBM CONTROLLER BOARD

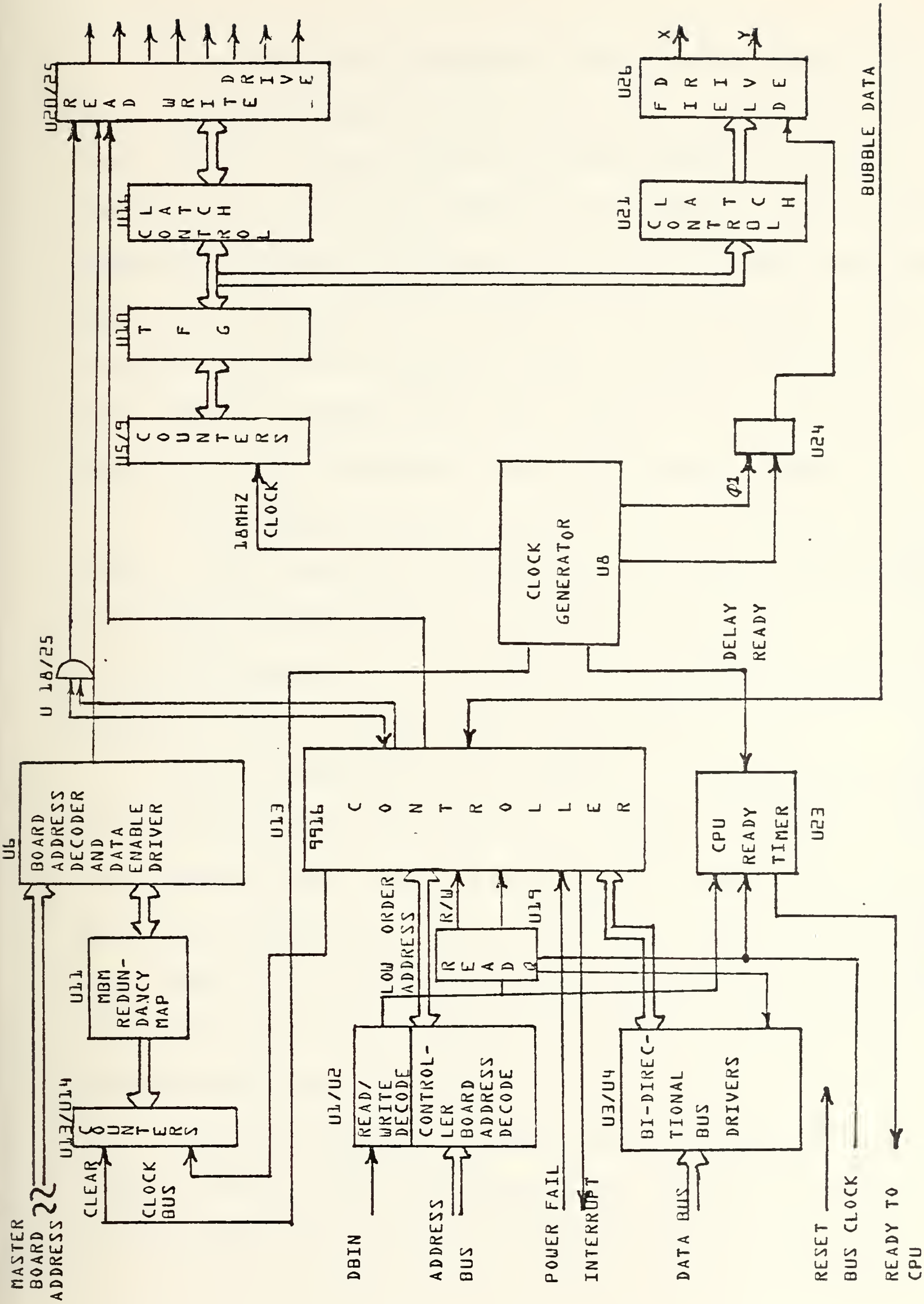


FIGURE 11 MBM CONTROLLER BLOCK DIAGRAM

The system 80/20 guarantees 5.3 ms of warning prior to the loss of +5V. The MBM chip requires 6.41 ms for a worst case return of all minor loops to zero. A storage capacitor may be required to meet this 1.31 ms time difference. Data Bus In (DBIN) is the controller device Read/Write command and is derived from Memory Read (MRDC/) or Memory Write Command (MWRC/) on the master bus. The controller uses this command along with the other two busses to interpret commands from the microcomputer. Memory Enable (MEMEN/) is simple a low logic enabling signal acting much like a chip select. It is discussed in the board addressing scheme.

The primary job of the controller board is Read/Write loop control. Major functions of the board, and the associated components, are: Timing (U8, U19, U23, U24), Data Bus Interface (U3, U4), Addressing (U1, U2, U19), Redundancy (U6, U4, U14), Timing Function Generation (U5, U9, U10, U16, U21) and Bubble Board Interface (U20, 25, 26).

U8 is a conventional clock generator. It provides clock reference within the control and bubble boards. As mentioned earlier, CPU interface timing is accomplished with BCLK/ and associated ready timing of U23B. Board sync is generated by clocking the inputs to U23A. U19 provides timing gates to the Timing Function Generator (TFG) circuitry as well as clocking the input data. U24 supplies enable timing to the bubble module for coil start/stop timing.

Data bus interface is accomplished with two Bidirectional Bus Drivers, U3 and U4. With an enabling input of Ready from U13, the read or write signals from U19B or U1 determine direction of data flow.

The controller board is addressed as a memory mapped device utilizing 15 address lines. SBC address 14 and 15 were "ANDed" together on the interface board to reduce the number of SBC address lines to 15 to conform to the controller address bus, mapping the controller into high memory. The memory mapped address is determined by the controller (A0-A3), U1 (A4-6) and U2 (A7-A14 & 15). U1 and U2 are fusable link ROMs (256x4). U2 contains a hex "0" at its address 0FFh. U1 contains a hex "E" at its address "47h" and "8" at address 4F. In addition to the address bus, U1 utilizes a high level logic memory enable (MEMEN) signal created from "OR"ing SBC memory read (MRDC/) with SBC memory write (MWRC/) on the interface board. Finally, the actual address selection (within U1) is determined by the Data Bus In (DBIN) signal. For DBIN true (high), U1 address 4Fh is accessed, which generates a chip select and read signal to the controller board. For DBIN false, address 47h is accessed, which generates a write and chip select signal to the controller board. The result up to this point is that for an address of 0FFFXh (where X is a don't care) the controller can differentiate between a read and a write memory operation.

The lower four address bits are fed directly to the controller. All controller commands are mapped onto these address locations (0FFF0h to FFFEh) with 0FFh not utilized. TMS 9916/5502 controller specifications [Reference (5)] details the commands. All commands with the exception of 0FFF2h utilize the data bus to pass required data associated with the command. The 0FFF2h command (Control Command) is an expansion command that allows the data bus to be utilized to pass eight additional commands.

These commands have no data associated with them, so no conflict occurs. For example, OFFF2h on the address bus, with 02h on the data bus causes the controller to execute its microprogram to read a page of bubble data into the controller buffer, reposition required loops, etc. It should be noted that control commands are transparent to the programmer, in that once the controller is programmed, the CPU may go on and leave the controller to complete its task. Monitoring may either be done via status polling or interrupt programming. This will be discussed further in Section III.

Timing interface for read/write control timing is accomplished via the CPU ready circuit. If the controller required additional time in accessing its buffers to the bus, it may pull the ready line low, causing the CPU to enter a wait state until the controller is once again ready.

Redundancy circuit discussion follows. Starting after a reset generated by the read Counter Clear (RDCTRCL) signal from the controller (U13), U14 counts the gated 50KHZ clock. Its output is fed to the 256X4 bit Schottky fusible link ROM, U11. This ROM contains the redundancy map for up to four installed MBM chips and is the same type as the previously discussed U1 and U2. The current application has only one MBM, so U11 contains only one of a possible four maps. Table I is the redundancy map for the MBM employed in this report. The counted clock input causes the redundancy map to be output to U6, which generates a Data Enable (DATAEN) signal for each valid minor loop, and fails to post the DATAEN signal for bad loops, corresponding to the ROM map. This signal

TABLE I

REDUNDANCY MAP

```
     FINAL MASK
MODULE   89-92-10     23 June 77     13:43:35     TEMP 0 CENTIGRADE
 0000 0002 01C0 0010  0000 0000 0000 00000000 0007


BAD LOOP ADDRESSES (HEXIDECIMAL)
 001E, 0027, 0028, 0029, 0038
```

is gated with the controller Data Out signal to form the Good Data (GDATA) signal at U25D. The signal is then transferred to the MBM via buffer U25 to gate out bad loop information.

U6 receives three enable lines, Board Select (BDSEL) A, B, and C. These are used to indicate which of up to eight bubble board redundancy maps are to be accessed. Currently the select lines are tied to ground to supply a logic zero to the board address logic. In a larger application with more than four bubble boards, an additional redundancy map ROM would be added, generating inputs to U6 (D4-7), and would contain the maps for boards 4-7. Additionally, active board address would have to be supplied to the board select lines.

Precision timing waveforms for all bubble board functions are generated in the Timing Function Generator group. U10, the Timing Function Generator (TFG), is another fusible link ROM. The contents of this ROM are indicated in Table II. Counters U5 and U9, driven by the clock generator 18 MHZ output, and under control of U12, access the TFG. It in turn generates output signals on its data lines to sequence a set of timing pulses to two eight input, "D" flip flop latches, U16 and U21. U21 utilizes only six of its eight "D" flip flops. This latched sequence is clocked by the counted down 18 MHZ clock (4.5 MHZ) to generate output signals to drivers U20, U25 and U26. These three chips are quad, two-input NOR gates and make up the function drivers. All MBM Board commands serve as inputs to these drivers to be timed under micro sequence control of the TFG.

The interrupt output from the controller allows for a powerful interface to the microcomputer. When enabled via its mask register, interrupt is generated from U1/34 through driver U13A to B2/39. Software is written for this to go in as interrupt five to the SBC; however, it is not currently implemented.

TABLE II

EVALUATION CARD FUNCTION TIMING PROM CONTENTS ('S471)

ADDRESS BASE	EVEN LOCATIONS (functions)								ODD LOCATIONS (drive pulses)							
	0	2	4	6	8	A	C	E	1	3	5	7	9	B	D	F
020	X	X	X	00	00	10	10	00	X	X	X	06	06	04	04	04
030	01	01	01	01	01	01	01	01	04	04	04	04	04	04	04	08
040	00	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08
050	0A	0A	0E	0E	0E	0E	2E	2E	08	09	09	01	01	01	01	01
060	2E	6A	2A	28	28	28	20	00	01	01	01	01	01	03	02	02
070	00	00	00	00	00	00	00	00	02	02	02	02	02	02	02	02

OUTPUT	FUNCTION
1	XOUT
2	XIN
3	REP
4	ANH
5	GEN
6	CLP
7	STB
8	-

A	DRIVE	(X+)
B	DRIVE	(Y+)
C	DRIVE	(X-)
D	DRIVE	(Y-)

X = DON'T CARE

III. CAPABILITY EXPANSION

A. INTERRUPT VERSUS STATUS CHECK

The system designed in this thesis can be operated in one of two ways. One is for the Central Processing Unit (CPU) to obtain input data when available, examine it, store as required, waiting for the mass memory system to complete its operation (a significant amount of time), and then wait for the next input cycle (an even more significant delay). Due to the speed of the microprocessor, the relatively infrequent occurrence of input data, and the additional delay involved with a relatively slow mass memory, the CPU would spend most of its time "polling" the input or output, waiting for either an input operation, or for the memory to complete its operation. This is trivial, if the CPU has no other task; however, if it could be gainfully employed elsewhere, it represents a great waste of computing power.

An alternative way of accomplishing the same task would be for the computer to be working continuously. When the input data bus brought information to the system, it would interrupt it. The microcomputer could then accept data from the 1553 bus buffer, operate on it, and store as required. Utilizing an MBM controller that can be told to take the data, store it and generate an interrupt when complete, the MBM module could be left to its work, and the microcomputer returned to the task interrupted.

For a Metal Nitride Oxide Semiconductor (NNOS) nonvolatile mass memory, or other type of slow mass memory without a separate intelligent controller, the same capability may be realized through software, utilizing

the programmable interrupt timer of the SBC. By setting this timer to the required memory delay, an interrupt would be generated at memory completion.

Utilizing this method based on interrupts, the recorder system could realize a ten-fold increase in computing power. This power could be well utilized in the present and future monitoring applications.

B. SAFETY PROGRAMMING

A data recorder, by definition, has considerable information available to it. In the case of an aircraft crash data recorder, a wealth of aircraft status data is sent each second to the recorder. For a microcomputer based system linked to a MIL STD 1553 data bus, with other aircraft systems also on the bus, the state information available is sufficient for many safety calculations.

Reference [2] discusses the use of discrete parameters to represent a wealth of aircraft status data in a very compact form. The variable DP&1 and DP&2 of the RECORD program represent 16 BIT words composed of discrete data. A suggested implementation is shown in Table IV, where DP&1 represents pilot input, and DP&2 represents aircraft state. Utilizing this type of data, the recorder has available to it information on flight perturbation, pilot response, control response resulting, and finally the aircraft response resulting from this chain.

With aircraft air speed, fuel load, position, etc., available, calculations of fuel exhaustion time and position, optimum climb, cruise, and loiter configuration, or constant energy display mapping could be supported. Real time calculation of take-off time and distance could be automatically calculated, with no pilot generated input data other than

field length. Real time warnings could be output, if the field length is insufficient for take-off, as a function of real time sensing of configuration and ambient conditions.

These are examples of applications for all aviation. A potentially more important task is related to the Navy carrier mission. By continuously having available to it such a wide variety of parameters, and due to its near instantaneous analysis capability, the microcomputer could be programmed to recognize certain definable "extremis" situations at the very earliest stages of their onset. Through this recognition algorithm an "eject alert" could be generated, allowing for immediate pilot correction and/or additional time to analyse the critical eject decision. In the familiar dark night launch, which is a time-critical situation, this could very easily make the difference between successful recovery/ejection or aircraft and pilot loss.

The RECORD program of Appendix B is interrupt driven at its outer level. The loop that comprises the main body of this program simulates a calculation of the type described. This loop calls the Eject Alert procedure if the parameters evaluated indicate that a critical situation has developed. In the example there is little doubt that an "extremis" situation exists. An extreme example was chosen to demonstrate that a computer can recognize specific situations if:

1. it has sufficient status data;
2. it is programmed to recognize these input values as a set that correspond to a critical situation.

The programmed example assumes arbitrarily that discrete data words DP$1 and DP$2 are implemented as shown in Table IV. Based on this, a

value of 1E83h would indicate a very dire situation. The aircraft would have fully split flaps, fully split slats, wide open speed brakes and hung gear. Simultaneous checks of altitude through the program indicate that the aircraft is below 100 feet and falling. This is an example of an easily defined "extremis" situation.

The program calls the Eject Alert procedure to provide warning. The significance is that the microcomputer can analyse the parameter each second, detect a situation such as this, and furnish a warning long before reaching this point. Placing the routines in the main body of the program insures that the parameters examined are at most one second old, since the 1553 bus would give new data each second.

IV. RESULTS AND RECOMMENDATIONS

The complete system was assembled after considerable delay in obtaining the various component parts. Some difficulty was experienced with the CPU coming loose from its socket on one side due to board flexure. Consideration might be given to soldering the CPU into the socket prior to flight testing.

Test software was designed initially to repetitively read or write to the controller, until the controller buffer was full, then to transfer the page to the bubble module. This program was repetitively looped while signal checks were conducted. It was determined that the test oscilloscope utilized was not sufficiently fast to syncronize and display the waveforms of interest.

Test software was altered to verify operation of the 9916 First In-First Out (FIFO) buffer, by executive a write of 17 bytes after system initiation, followed by a read of the FIFO. This was also unsuccessful.

Consultation with Texas Instruments personnel indicated several changes to be incorporated, as indicated in Appendix C. Additionally, the 9916 controller performance may possibly be temperature dependent to a greater degree than listed in its specifications. Cooling air was supplied for future tests, but testing was not resumed in time to see if this solved the temperature problem.

Due to changes required, testing was not completed and remains as the final task. Subsequent work should include correction of deficiencies listed in Appendix C, and check out of the address ROMs. The Controller

operation should be verified independent of the MBM board. It is recommended that a status polled program be utilized, as recent discussions with TI personnel indicate that the interrupt routein from the controller may operate in variance with the specifications (Reference [5]).

Research should continue with the MBM. It is the best medium for the mission, and future technology growth will only accentuate this. Consideration should be given to simultaneous development of an MNOS based system utilizing the same computer, as discussed in the thesis. The technology risk here is low; however, the storage density is also considerably lower.

Finally, the interface design to a data bus which serves to deliver the status information needs to be completed.

APPENDIX A

BUBBLE TECHNOLOGY

A. INTRODUCTION

In many materials there exist "domains" of magnetization. These domains are usually randomly aligned, such that the net magnetization (magnetic energy) of the material is near zero. In certain materials, a large number of these domains align along some axis with their magnetic poles in the same direction. This is a naturally magnetic material. In an artificially induced magnetic material the same result occurs; however, the alignment is forced by an outside electromagnetic (H) field.

If a naturally magnetic material is placed in an H field aligned with its principal magnetic axis, all domains will tend to align with this field. Those domains that were aligned opposite to the field will be reduced in size as a function of their magnetic dipole moment, their initial polar direction, and the strength of the external magnetic field (bias). By careful selection of the magnetic substrate utilized, and application of the proper bias, those domains in opposition to the bias field can be caused to reduce in size until they are arbitrarily small "bubbles" of polarized material within a "sea" of oppositely polarized material (Reference [6]). Variation of bias field and material properties of the substrate will determine bubble size. If the bias field is allowed to become too strong, they will be annihilated, i.e., caused to collapse into the "sea." If too weak, the bubbles will be too large, with resulting propagation and storage problems.

As a first step toward usability, the bubble must be caused to move under control. For this purpose a combination of "tracks" is laid down in the substrate, usually from permalloy material. An external electromagnetic field is applied (two periodic signals, 90° out of phase), such that it induces a magnetic field within the plane of the substrate. The moving force is caused by variations in flux density, due to the permeability of the permalloy pattern, that causes the tracks to develop magnetic poles, resulting in the bubbles being moved along the track in the direction determined by the external rotating field. The bubble moves in the direction of reduced bias, at a speed proportional to the difference between the non-uniform bias across the bubble diameter and the coercivity (Reference [6]). The bubble movement is accomplished by realignment of the magnetic vectors at successive locations within the substrate.

The shape of the permalloy track elements have a distinct effect on propagation speed and reliability. Patterns used in the past have included a chevron, a "T" bar, and a crescent. The bubble chip used in this thesis employs the "T" bar pattern. Current research in higher density chips is employing an asymetrical chevron to achieve higher packing densities and greater field rotation rates.

Three other basic functions are required to make up a useful bubble storage device. Information must be written into the device (bubble generation). The information must be read from the device (bubble detection) and bubbles must be deleted from the device (annihilation). Generation can be accomplished via a fine current loop which, at a specific

point in the bubble track, can be pulsed in opposition to the static bias field to produce a bubble. Similarly, annihilation can be accomplished by bringing the bubble under the same loop and reversing the direction of current flow. For data handling, the generate and annihilate functions are usually separate. Bubble detection can be accomplished in several ways. One is to cause the bubble to be stretched out, and then run under a permalloy magnetorestive detector. The change in the resistance of the detector due to the field change induced at bubble passage can be detected and amplified as the module output. Interaction of the rotating field and the detector is handled by putting the sense element in one leg of a balanced bridge network, with the other legs in the rotating field but not exposed to bubble passage.

Physical arrangement of the permalloy tracks determine the usefulness of the memory. All bubble positions and functions could be arranged around a loop; however, as total storage increased, access time would go up linearly, just as in serial magnetic tape systems. A more practical approach for systems that require random access is to model the system after a fixed head disc, in which information is fed to several heads simultaneously, transferred to the disc tracks, and then the disc rotated to the next data position. This is the model for the major/minor loop MBM chip layout.

This design requires the additional bubble function of "transfer," which moves a bubble from/to the major and minor loops. This propagation directionality is obtained by application of very carefully timed signals to the transfer gates themselves to cause the propagation vector to move

toward the minor loops, via specially shaped permalloy track elements, at the instant the bubble is positioned at the gates. In all bubble functions, element size, placement, and spacing are critical.

B. CONSTRUCTION

Solid state construction techniques very similar to that used in the fabrication of other microelectronic components are utilized in the construction of the bubble memory. This contributes greatly to the reliability and low cost of these units. The current MBM utilized is a production version of approximately 100K bits. Texas Instruments expects to market a 256K bit chip in the same package by mid-1978. Four of these larger capacity chips will be mounted, with drive circuitry, on the same board utilized in the current application, to yield a one megabit storage system.

One area of concern in bubble chip reliability is bias field variation susceptibility (Reference [11]). Element differences within the device, as well as variations of the field strength of the bias magnets, may cause the bias margin to be unacceptably small. In this case relatively small increases in flux density may cause bubble annihilation, with resulting data loss. On the other hand, a weaker field may cause overly large bubbles, with attendant strip out problems. Bias field margin is computer tested. A final test is a go/no-go check for the completed module. As the bubble chip capacity is increased by reducing the bubble size and spacing, this will be an area to watch. Research into permalloy track element shape is producing patterns that require less precise manufacturing technique while yielding a wider bias margin (Reference [12]).

Another area of possible problems arises with the major/minor loop architecture. To obtain an acceptable yield with a chip employing extremely close manufacturing tolerances for the function elements, a failure margin must be allowed. Keenan and Naden (Reference [14]) report that for the TBM 0101 chip, up to 13 of the 157 minor loops are allowed to be defective to obtain desired yields. The actual map of the assembled chip is obtained in final assembly testing, with bad minor loops noted. It then becomes the job of the bubble controller unit to selectively skip these loops on read/write transfer operations. An additional caution must be observed with regard to this redundancy mask in that if not correct, and if bubbles are transferred into the bad minor loops, recurring problems with erroneous data may result thereafter. D. M. Lee (Reference [6]) outlines the method of employing the mask information to gate out bad loops. He further outlines the entire controller circuitry required for MBM control.

C. APPLICATIONS

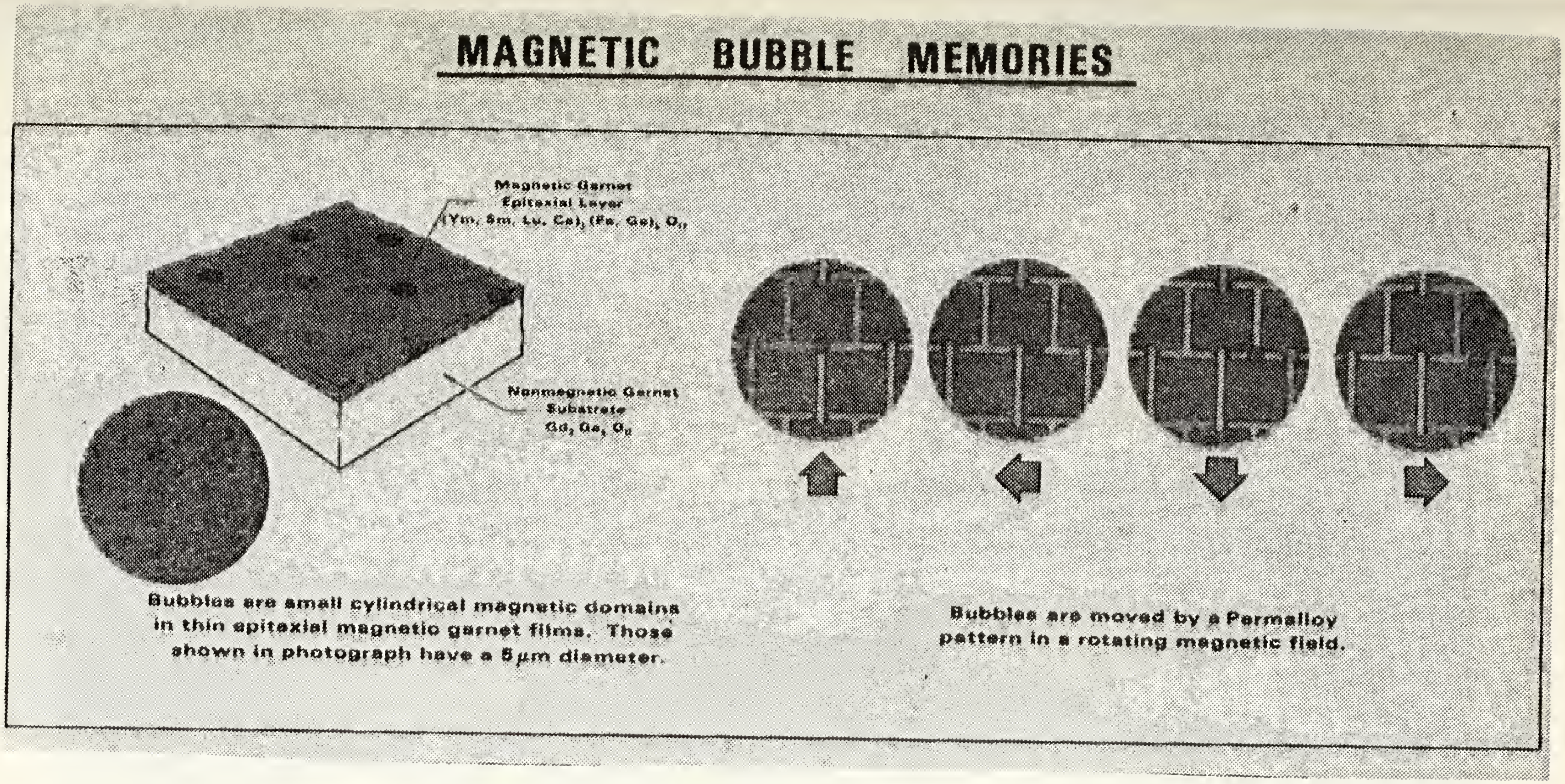
NASA and DOD are both currently funding research into magnetic bubble storage devices. Rockwell is building a 10^8 bit space qualified recording system for NASA (Reference [13]) employing signle loop architecture. It is hoped to replace three mechanically oriented systems with the one bubble system in future applications to decrease weight, improve reliability and reduce power consumption.

Texas Instruments, under contract to the Air Force Avionics Laboratory (AFAL), is developing second generation MBM modules in 256K-1M bits/chip

range, utilizing the major/minor loop architecture. Specific applications of the AFAL work are not yet indicated.

Bubble drive circuitry is passive, except during actual memory access. Function circuitry is similarly inactive, except when actually performing the intended function. There are no quiescent bias currents needed in a stand-by mode. For this reason, MBM will be applied to many applications where power consumption is a consideration. By actively switching all bubble functions, minimal power drain may be realized.

Magnetic bubble memories will find application wherever the low cose/bit and non-volatile nature are important, and where the relatively longer access time required can be tolerated.



MAGNETIC BUBBLE MEMORIES

Bubbles are small cylindrical magnetic domains in thin epitaxial magnetic garnet films. Those shown in photograph have a 5 μm diameter.

Bubbles are moved by a Permalloy pattern in a rotating magnetic field.

FIGURE 12

MAGNETIC BUBBLES AND TRACKS

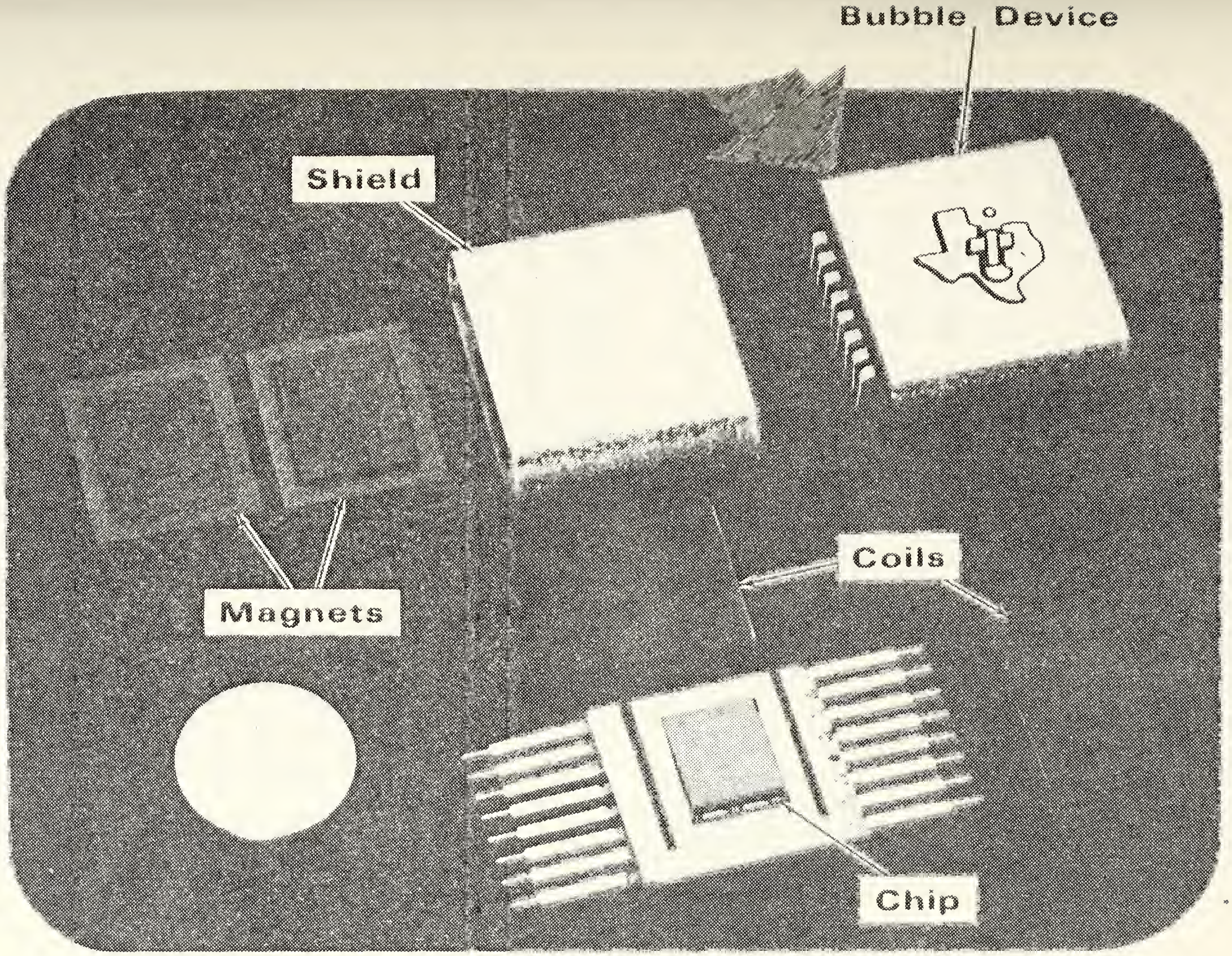


FIGURE 13

TBM 0101

TABLE III

TMS 9916/TBM 0101 PARAMETERS

TOTAL STORAGE	100,637 BITS
USABLE STORAGE	92,160 BITS
NUMBER OF MAJOR LOOPS	1
NUMBER OF MINOR LOOPS	157
GUARANTEED NUMBER OF USABLE MINOR LOOPS	144
MAJOR LOOP LENGTH	640
MINOR LOOP LENGTH	641
PAGE SIZE	18 BYTES
NUMBER OF PAGES	641
SINGLE PAGE MODE MAXIMUM PAGE WRITE TIME	12.82 MS
MULTI-PAGE MODE AVERAGE WRITE TIME	3.22 MS
FIRST BYTE AVERAGE ACCESS TIME	6.41 MS
MINIMUM SHUT DOWN TIME	6.41 MS

APPENDIX B

COMPUTER PROGRAMS

A. GENERAL

Development support for both hardware and software is available in the Intel MDS 800 Microcomputer Develop-System. Figure 14 illustrates the entire development system. Utilizing the In Circuit Emulator (ICE), SBC hardware can be simulated to a great degree. The ICE is currently incapable of simulating interrupt driven routines for the SBC, as the interrupt controller cannot be simulated. A modification to allow simulation of SBC interrupt structure is available and Intel has been contacted to obtain it.

Due to the memory mapping of the ICOM PROM programmer board, PROM programming within the ISIS operating system is not straight forward. The ICOM PROM programmer is mapped into the top 16K of memory and as such is not compatible with a system configured for more than 48K of memory. Dip switches on the board would allow the board to be re-addressed, but it would then be incompatible with its own monitor.

For 62K CPM, the debugged program is loaded into a user RAM area below 48K and run. Upon exit to the monitor the top 16K RAM board is removed with the system on, and the ICOM programmer board inserted. The system is then rebooted on the monitor and program control transferred to the programmer (Reference [4]). For PLM-80, and other languages run under the 64K Intel System Implementation Supervisor (ISIS), another problem arises. Due to SBC EPROM memory mapping, the SBC is mapped in ISIS resident area, and as such must have its operating load map transformed utilizing the ICE80.

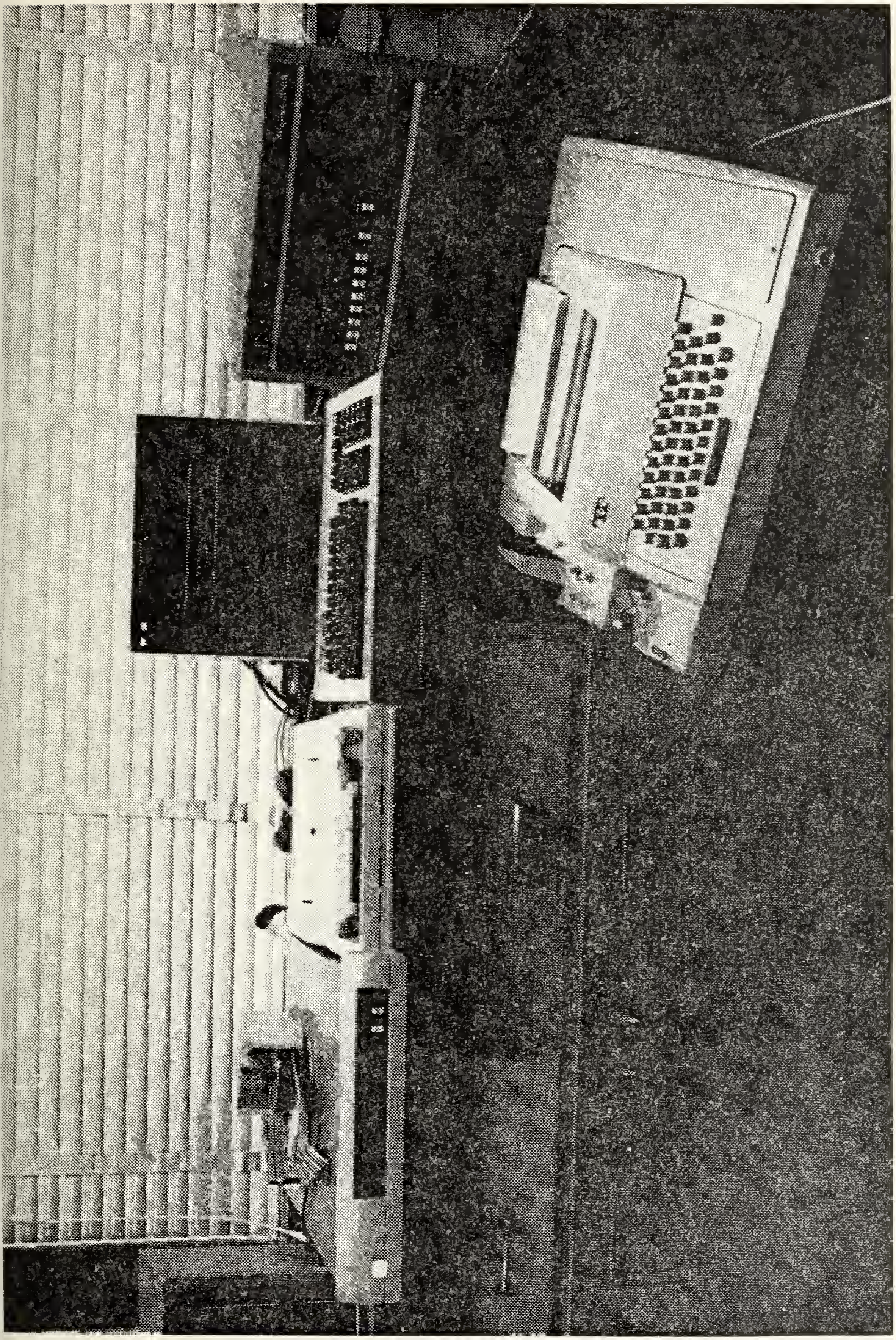


FIGURE 14

DEVELOPMENT SYSTEM

With the debugged program compiled, linked and located on disc, the ICE 80 is utilized to transform the program load map into user accessible RAM under the MDS/ICE 80 64K memory map. Care is taken not to transform into the top 16K of memory, as this will be removed to utilize the programmer. With the transformed program loaded under ICE 80, the disc door is opened and the system booted on the monitor. The remaining procedures are the same as for CP/M.

The decision on whether to use I/O ports or to memory map the controller was a difficult one. At the time the available MDS was not configured with a general purpose I/O board, so the decision was made to memory map the controller. Subsequently, the MDS 504 General Purpose I/O Module has been obtained and is installed on the MDS. For the testing phase, consideration might be given to utilizing the controller, port mapped, due to the greater ease of signal checking. The required signal complementation of address and control busses could then be accomplished in software also.

Programs to accomplish the recording function as well as to test bubble module operation were developed. The essential elements of the PROCESS PLM program of Reference [2] were rewritten in PLM-80 as the RECORD program. The RECORD program incorporates interrupt initiation, MBM interface, and examples of real time analysis of input parameters to dynamically vary the compression.

The MEMORY RECORDING PROCEDURE of Reference [2] was rewritten on the MDS and designed to record into MNOS. If this program were adapted for the SBC port numbers, and expanded to handle a larger number of

MNOS chips, it could provide a demonstration data recorder while the bubble technology matures. The programs were not included, but are available if this route is chosen.

PLM-80 bubble driver routines were originally developed on the MDS; however, they could not be located at SBC EPROM addresses without ICE 80. The status-polled MBM drivers were written in assembly language, utilizing the Digital Research CP/M operating system. This allowed programming of EPROMS for testing. This requirement was removed when the ICE 80 was obtained, but time did not allow for program rewrite. This program translates directly into PLM-80, and if used in further work should be rewritten to aid documentation.

Development of the interrupt driven MBM Driver was accomplished in assembly language as well, to facilitate register and stack operations associated with interrupts. Translation of this program into PLM-80 is not as direct. Utilizing the PLM Stack pointer (STACKPTR) functions, a based variable would be utilized to store the return point stack pointers of each routine, and appropriate stack operations would allow movement back and forth between the "Outer Level" and the page write and end check routines as required.

B. RECORD PROGRAM

The RECORD program is the executive program. It performs three functions: real time analysis, parameter analysis and compression, and call control for the BUBBLE program.

1. Real Time Analysis

With the recording function written as an interrupt activated procedure, the software is free to perform real time analysis of

desired parameters as discussed in Section III. The sample calculation consisting of the IF statement in the "Outer Level" infinite loop is an example of this.

2. Parameter Analysis and Compression

The INPUT 1 procedure is interrupt driven. It in turn calls all other procedures with the exception of Eject Alert. Two examples of how to dynamically vary the data compression rate are included in the VERTG procedure and the ALTF analysis. In each case, the compression parameter, i.e., the allowable difference between old and new values, is adjusted dynamically as a function of data from one second ago.

3 MBM Call

After data has been analysed for changes and labeled, it is stored in an output buffer (one byte) and control is passed to the MBM driver.

C. BUBBLE DRIVER PROGRAM

This assembly language program is the MBM driver. It follows the flow chart of Figures 15 and 16. Figure 15 shows the sequence (Reference [5]) to initialize the controller at power up. Figure 16 is the single page read/write flow chart (Reference [5]). It does not use interrupts to control operations as it was developed from a version used for testing.

The various console calls at the input to the procedures are for the purpose of program debuging and testing and would be eliminated in the final version.

D. INTERRUPT CONTROLLED BUBBLE TEST PROGRAM

This program was developed from an original status polled test program to make the recording process more efficient. Translated to PLM-80 and linked to the RECORD program, it would allow the system to be used as a monitor/recorder.

The basic flow chart is the same as that of Figure 16, with the exception that the process does not wait for the controller to finish. Once the controller is told to write a page, the program returns directly to the "Outer Level" while the controller completes its task. When the controller is done, the generated interrupt five causes the program to leave the "Outer Level" and return to the End Check (ENDCK) routine via the Return Point (RTNPT). At completion, execution once again returns to the outer level.

The program was located in EPROM, but transferred into SBC RAM thus allowing program alterations during operation. To facilitate this, multiple no-op (NOP) instructions were inserted in each procedure, which permitted patching modifications. The trace routines are similar in both MBM programs. The program is relocated on the SBC to its 3000h start address using the SBC monitor move (M) command prior to execution.

A read routine was incorporated into this program as well. It was derived from the original version of the test program, and as such is not interrupt controlled. This program is the main test program. In the testing mode, keyboard control of read/write operation is available and selection of the number of pages read/written is accomplished via program modification.

Tables V and VI are the memory maps of the SBC and MDS utilized.

```
      /*                      RECORD PROGRAM                  */


    /*  INPUT DATA IS PLACED IN AN INPUT BUFFER(UP TO 24 BYTES(12H))
        STARTING AT 3EDCH, UTILIZING THE MONITOR AND SILENT 700.
        THIS SIMULATES THE MIL STD 1553A  INTERFACE BUFFER.  NO
        ASCII DECODE ROUTINE IS UTILIZED AS THE MONITOR HANDLES
        THAT.  ONCE THE DATA IS IN RAM THE PROGRAM IS EXECUTED.
        EACH TIME AN INTERRUPT 6 IS GENERATED IT SIMULATES THE
        1553 INTERFACE INTERRUPTING WITH A FULL BUFFER OF NEW INPUT
        INFORMATION. THIS IS PROCESSED AND CONTROL RETURNED TO THE
        OUTER LEVEL BETWEEN INTERRUPTS.   */

RECORDER:
  DO;

DECLARE DCL LITERALLY 'DECLARE', LIT LITERALLY 'LITERALLY';
DCL TRUE LIT '0FFH', FALSE LIT '0', FOREVER LIT 'WHILE TRUE';
DCL (LIMIT2,LIMIT4,LIMIT6,LIMIT7) BYTE;
DCL (J,MINFLAG,SECFLAG) BYTE;
DCL (K1,OLDMIN,NEWMIN,NEWSEC) ADDRESS;
DCL (ALTC,ALTF,KCAS,HEAD,DPS1,DPS2,VG) ADDRESS;
DCL INPUT$BASE ADDRESS AT (3EDCH); /* INPUT BUFFER AT 3000H */
DCL (INPUTBUFF BASED INPUT$BASE) (24) ADDRESS;
DCL STALL BYTE DATA(150);
DCL BUBBLE ADDRESS AT (1000H);
DCL DATTA BYTE AT (3F00H);
DCL FLAG1 BYTE AT (3F01H);
DCL EOIC LIT '20H';   /* END OF INTERRUPT COMMAND */
DCL INT5VECTOR ADDRESS AT (3FF5H);
DCL INT6VECTOR ADDRESS AT (3FF9H);
DCL ICCP LIT '0DAH';   /*   INTERRUPT COMMAND PORT ADDRESS   */
DCL LOW1 LIT '4000', LOW2 LIT '100';
DCL SLOW LIT '120'; DEEP$TROUBLE LIT '1E63H', FASTSSINK LIT '10';


          /*             PROGRAM PROCEDURES              */

CONVERT:
  PROCEDURE (VALUE) ADDRESS;  /* REVERSES BYTE ORDER TO ACCOMODATE
      THE 8080  */

    DCL VALUE ADDRESS;
    RETURN SHL(VALUE,8) + HIGH(VALUE);
  END CONVERT;

NEXTPARAM:
  PROCEDURE ADDRESS; /* FETCHES THE NEXT PARAMETER FROM THE BUFFER */
*
```

```
  DCL ITEM ADDRESS;
   ITEM=INPUTBUFF(J);
   J=J+1;
   RETURN CONVERT(ITEM);
 END NEXTPARAM;


RECORD:
 PROCEDURE (ITEM, NAME);
   DCL ITEM ADDRESS;
   DCL NAME BYTE;

     IF MINFLAG THEN
              DO;
         DATTA=HIGH(CONVERT(NEWMIN)); /* LABEL INVISIBLE(0),
                                             WRITE TO FIFO */
         CALL BUBBLE;

         DATTA=LOW(CONVERT(NEWMIN));
         CALL BUBBLE;
               MINFLAG=FALSE;
              END;

     IF SECFLAG THEN
              DO;
             DATTA=HIGH(CONVERT(NEWSEC+1000H)); /* ADD LABEL(1) */
             CALL BUBBLE;

             DATTA=LOW(CONVERT(NEWSEC));
             CALL BUBBLE;

               SECFLAG=FALSE;
              END;

       DATTA=HIGH(CONVERT(ITEM+SHL(DOUBLE(NAME),12))); /* ADD LABEL
    IN HIGH ORDER 1/2 BYTE  */
    CALL BUBBLE;

       DATTA=LOW(CONVERT(ITEM+SHL(DOUBLE(NAME),12)));

    CALL BUBBLE;
 RETURN;
END RECORD;

ALGO:
 PROCEDURE (OLD,THRESH,NAME) ADDRESS;
   DCL (OLD,NEW,DIFF) ADDRESS;
   DCL(THRESH,NAME) BYTE;
     NEW=NEXTPARAM;
     IF NEW > OLD THEN DIFF=NEW-OLD;
*
```

```
                   ELSE DIFF=OLD-NEW;

      IF DIFF > DOUBLE(THRESH) THEN
                             DO;
                           CALL RECORD(NEW,NAME);
                           RETURN NEW;
                           END;

                      ELSE RETURN OLD;

END ALGO;

VERTG:
  PROCEDURE (NAME);
    DCL NAME BYTE;

      IF((VG>3) OR (DPS2 AND 0006H))
         THEN LIMIT2=0;  /* SINK RATE CHECK. COMPRESSION PARAMETER
                         ADJUSTED IF A CARRIER OR FCLP LANDING    */

         ELSE LIMIT2=1;  /*  DEFAULT VALUE   */


      VG=ALGO(VG,LIMIT2,NAME);
    RETURN;
  END VERTG;

EJECTSALERT:
  PROCEDURE;
    /*  THIS DUMMY PROCEDURE WOULD FURNISH EJECT WARNING IN THE FORM
        OF A LIGHT, AURAL TONE, OR OTHER DESIRED FORM.  AN ALTER-
        NATIVE WOULD BE ACTUAL AUTOMATIC EJECTION.     */

    RETURN;
  END EJECTSALERT;


INPUT1:
  PROCEDURE INTERRUPT 6;

     ENABLE;  /*  ENABLE INTERRUPTS WITHIN PROCEDURE   */

    /*  SCALE OF ALL INPUTS IS 1:1, IE '0FFH' FOR VELOCITY(KCAS)
    =255KTS.  ALTC GOES IN 4000 FT STEPS.   */

    J=0; /* RESET COUNTERS */
    NEWMIN=NEXTPARAM;

     IF OLDMIN <> NEWMIN THEN
           DO;
*
```

```
      OLDMIN=NEWMIN;
           MINFLAG=TRUE;
         END;

      NEWSEC=NEXTPARAM;
      SECFLAG=TRUE;

      CALL VERTG(2);
      ALTC=ALGO(ALTC,0,3);

   /* ADJUSTMENT OF FINE ALTITUDE COMPRESSION PARAMETER ACCORDING TO
      LAST ALTITUDE, AIRSPEED, GEAR POSITION.   */

   IF(((ALTF < 1000) OR (KCAS < (STALL + 5)))
     OR ((DPS1 AND 0070H) < 0070H))
         THEN LIMIT4=10;
         ELSE LIMIT4=100;

          ALTF=ALGO(ALTF,LIMIT4,4);
          CALL VERTG(5);
          KCAS=ALGO(KCAS,LIMIT6,6);
          HEAD=ALGO(HEAD,LIMIT7,7);
          CALL VERTG(8);
          DPS1=ALGO(DPS1,0,9);
          DPS2=ALGO(DPS2,0,10);
          CALL VERTG(11);

      DISABLE;   /*  DISABLE INTERRUPTS AROUND CALLS TO INTERRUPT
                               CONTROLLER  */
      OUTPUT(ICCP) = EOIC;  /*  OUTPUT THE END OF INTERRUPT   */
      ENABLE;               /*  RE-ENABLE INTERRUPTS   */

  RETURN;
 END INPUT1;


  STACKPTR=3F60H;  /*INITIALIZE PROGRAM STACK POINTER   */
  LIMIT2,LIMIT4,LIMIT6,LIMIT7=0;  /*  INITIALIZE LIMIT VALUES   */
  J,MINFLAG,SECFLAG=0;  /*  INITIALIZE FLAGS AND COUNTERS   */
  K1,OLDMIN=0;
  ALTC,ALTF,KCAS,HEAD,DPS1,DPS2,VG=0;  /*  INITIALIZE VARIABLES   */


FLAG1=FALSE;  /*  SET UP INITIALIZATION FLAG FOR BUBBLE CONTROLLER   */


DO FOREVER;
  INT6VECTOR = (.INPUT1); /*SET UP RAM INTERRUPT VECTOR TO CAUSE
       INTERRUPT 6 TO VECTOR TO INPUT1 PROCEDURE.*/

*
```

```
   IF((ALTC < LOW1) AND (ALTF < LOW2) AND (VG > FAST$SINK)
       AND (KCAS < SLOW) OR (DP$2 AND DEEP$TROUBLE)))
     THEN CALL EJECT$ALERT;

END; /* LOOP, WAITING FOR INTERRUPTS.  THIS LOOP IS THE 'OUTER LEVEL'.
       ANY NUMBER OF SAFETY OF FLIGHT ROUTINES COULD BE ACTIVE IN
      THIS AREA, RUNNING UNTIL INTERRUPTED, AND THEN RESUMED.
      ONE EXAMPLE IS INCLUDED OF A MACRO CHECK FOR AN EXTREME
      FLIGHT CONDITION.  SEE PROGRAM NOTES FOR EXPLANATION  */


END RECORDER;
```

```
;       BUBBLE DRIVER PROGRAM-STATUS POLLED


ORG 0C00H                ;START PROGRAM ON 4TH EPROM CHIP
                 ;TRACE ROUTINE MAP: *=BUBBLE. 2=INITIALIZE.
                 ;3=BUSY CHECK. 4=PAGEWRITE. 5=END CHECK.
                 ;9=LO BYTE CHECK. A=ZERO PAGE.

                 ;MAIN ROUTINE MUST INITIALIZE IFLAG TO ZERO
                 ;UPON PROGRAM RESTART. ADDITIONALLY, INPUT
                 ;BUBBLE DATA MUST BE PASSED VIA MEMORY INTO
                 ;LOCATION WRBUFF. STACK POINTER INTENTIONALY
                 ;NOT INITIALIZED.  TRACE ROUTINES FOR DEBUG.


VRBLS EQU 3F00H          ;VARIABLE STORAGE AREA

LDPGRGLO EQU 0FFF0H      ;LOAD PAGE SELECT REGISTER, LO BYTE
LDPGRGHI EQU 0FFF1H      ;LOAD PAGE SELECT REGISTER, HI BYTE
CONTCOM EQU 0FFF2H       ;READ CONTROL COMMANDS FROM DATA BUSS
RDBYTE EQU 0FFF3H        ;READ DATA BYTE FROM CONTROLLER FIFO
WRBYTE EQU 0FFF4H        ;WRITE DATA BYTE TO CONTROLLER FIFO
RDSTATUS EQU 0FFF5H      ;READ STATUS REGISTER
PGCNTLO EQU 0FFF6H       ;READ/WRITE NUMBER OF PAGES FOR MULTI
                         ;PAGE TRANSFERS, LO BYTE
PGCNTHI EQU 0FFF7H       ;SAME FOR HI BYTE
LDMINSZLO EQU 0FFF8H     ;LOAD LOW BYTE, MINOR LOOP SIZE(INIT-
                         ;IALIZATION ONLY)
LDMINSZHI EQU 0FFF9H     ;SAME FOR HI BYTE
PGPOSLO EQU 0FFFAH       ;PAGE POSITION COUNTER LOW BYTE. USED
                         ;TO SHOW CURRENT PAGE AT TRANSFER
                         ;GATE IN THE BUBBLE
PGPOSHI EQU 0FFFBH       ;SAME FOR HI BYTE
LDPGSZRG EQU 0FFFCH      ;LOAD PAGE SIZE REGISTER
                         ;(INITIALIZATION ONLY)
RDROMBYTE EQU 0FFFDH     ;READ CURRENT REDUNDANCY MAP,
                         ;INCREMENT POINTER
RWSTADDR EQU 0FFFEH      ;READ REDUNDANCY MAP ADDRESS

INITIAL EQU 1            ;INITIALIZE THE CONTROLLER
RDPG EQU 2               ;READ PAGE,SINGLE PAGE MODE
WRPG EQU 4               ;WRITE PAGE, SINGLE PAGE MODE
SGLPGMD EQU 8            ;SET SINGLE PAGE MODE
MULPGMD EQU 10H          ;SET MULTI-PAGE MODE
TESTMD EQU 20H           ;SET TEST MODE
RESET EQU 40H            ;SOFTWARE RESET
INTMSK EQU 80H           ;SET CONTROLLER INTERRUPT MASK

CO EQU 00FH              ;CONSOLE OUT CALL ADDRESS
```

```
        CI EQU 012H               ;CONSOLE IN CALL ADDRESS
        BUSY EQU 10H              ;DEFINE BUSY CHECK BYTE
        MINLPSIZ EQU 641D         ;NUMBER OF POSITIONS ON MINOR LOOP
        PAGESIZE EQU 18D          ;PAGE SIZE IN BYTES

BUBBLE: CALL INIT                 ;INITIALIZE IF RQD

        LHLD PAGENUM              ;LOAD UP DESIRED PAGE NUMBER
        XCHG                      ;
        LXI H,LDPGRGLO            ;LOAD UP COMMAND
        MOV A,E                   ;
        CMA                       ;INVERT DATA FOR BUSS
        MOV M,A                   ;OUTPUT LO BYTE OF PAGE
        INX H                     ;
        MOV A,D                   ;
        CMA                       ;INVERT DATA FOR BUSS
        MOV M,A                   ;OUTPUT HI BYTE COMMAND


        MVI C,'*'                 ;SET UP PATH TRACE
        CALL CO                   ;OUTPUT TRACE
        CALL BUSYCHK              ;SEE IF CONTROLLER DONE
        LXI H,WRBUFF              ;
        MOV A,M                   ;FETCH INPUT DATA FOR BUBBLE FIFO

        LXI H,WRBYTE              ;SET UP WRITE COMMAND
        CMA                       ;INVERT DATA FOR BUSS
        MOV M,A                   ;WRITE DATA BYTE TO FIFO
        LXI H,BYTECNT             ;SET UP BYTE COUNT INCREMENT
        INR M                     ;INCREMENT BYTE COUNT(INDIRECT)
        MVI A,PAGESIZE            ;
        CMP M                     ;
        CZ PGWRT                  ;IF BYTES WRITTEN = PAGE
                                  ;SIZE, WRITE PAGE
        RET                       ;RETURN HAVING WRITTEN DATA TO FIFO

PGWRT:  MVI C,'4'                 ;SET UP PATH TRACE
        CALL CO                   ;

        CALL BUSYCHK              ;SEE IF CONTROLLER BUSY
        LXI H,CONTCOM             ;
        MVI A,WRPG                ;
        CMA                       ;INVERT DATA FOR BUSS
        MOV M,A                   ;OUTPUT PAGE WRITE COMMAND TO
                                  ;CONTROLLER
        CALL BUSYCHK              ;CHECK FOR DONE
ENDCK:  MVI C, '5'                ;
        CALL CO                   ;OUTPUT TRACE

        LHLD PAGENUM              ;LOAD ADDRESS OF PAGE NUMBER
*
```

```
        INX H                   ;INCREMENT PAGE NUMBER INDIRECT

        SHLD PAGENUM            ;RESTORE INCREMENTED PAGE NUMBER
        XCHG                    ;PUT INCREMENTED PAGE NUMBER IN D/E
        LXI H,MINLPSIZ          ;CHECK FOR END OF MEMORY
        MOV A,H                 ;HI BYTE INTO ACCUM
        CMP D                   ;CHECK IF HI BYTE OF PAGE NUMBER =
                                ;HI BYTE OF MINOR LOOP SIZE
        CZ LOCHK                ;IF HI BYTES EQUAL, CHECK LO BYTES
        LXI H,BYTECNT           ;LOAD BYTE COUNT LOCATION INTO THE
                                ;H/L REGISTER
        MVI A,0                 ;
        MOV M,A                 ;RESET THE BYTE COUNT TO ZERO
        LXI D,PAGENUM           ;SET UP TO LOAD PAGE SELECT REGISTER
        LDAX D                  ;LOAD ACCUM FROM ADDRESS IN D/E
        LXI H,LDPGRGLO          ;LOAD H/L WITH LOAD PAGE REGISTER LO
                                ;COMMAND ADDRESS
        CMA                     ;INVERT DATA FOR BUSS
        MOV M,A                 ;OUTPUT LO BYTE TO PAGE SELECT REG
        INX H                   ;SET UP HI BYTE COMMAND
        INX D                   ;INCREMENT FOR HI BYTE
        LDAX D                  ;LOAD IN HI BYTE
        CMA                     ;INVERT DATA FOR BUSS
        MOV M,A                 ;OUTPUT HI BYTE TO PAGE SELECT REG
        RET                     ;

LOCHK:  MVI C,'9'               ;
        CALL CO                 ;OUTPUT TRACE

        MOV A,L                 ;
        CMP E                   ;CHECK LO BYTES
        CZ ZEROPG               ;RESET PAGE TO ZERO
                                ;IF AT END OF BUBBLE(LOOP MEMORY)
        RET                     ;

ZEROPG: MVI C,'A'               ;SET UP TRACE
        CALL CO                 ;

        LXI H,PAGENUM           ;LOAD H/L WITH ADDRESS OF PAGE NUMBER
        MVI A,00                ;ZERO ACCUMULATOR
        MOV M,A                 ;ZERO LO BYTE, PAGE NUMBER
        INX H                   ;INCREMENT
        MOV M,A                 ;ZERO HI BYTE

        RET                     ;


INIT:   MVI C,'2'               ;SET UP TRACE
        CALL CO                 ;

        LDA IFLAG               ;CHECK IF INITIALIZED ALREADY
*
```

```
        CPI 1                    ;SEE IF SET
        RZ                       ;IF SET, IE INITIALIZED, RETURN


        CALL BUSYCHK             ;
        MVI A,PAGESIZE           ;
        LXI H,LDPGSZRG           ;LOAD H/L WITH LOAD PAGE SIZE REG
                                 ;COMMAND
        CMA                      ;INVERT DATA FOR BUSS
        MOV M,A                  ;LOAD PAGE SIZE INTO PAGE SIZE REG
        LXI B,MINLPSIZ           ;
        MOV A,C                  ;LO BYTE, MINOR LOOP SIZE, INTO ACCUM
        LXI H,LDMINSZLO          ;LOAD H/L WITH LOAD MINOR LOOP SIZE
                                 ;LOW COMMAND ADDRESS
        CMA                      ;INVERT DATA FOR BUSS
        MOV M,A                  ;PUT MINOR SIZE LO BYTE IN MINOR LOOP
                                 ;LOOP SIZE REGISTER
        MOV A,B                  ;SET UP HI BYTE
        INX H                    ;
        CMA                      ;INVERT DATA FOR BUSS
        MOV M,A                  ;LOAD MINOR LOOP SIZE, HI BYTE
        MVI A,RESET              ;SET UP RESET COMMAND
        LXI H,CONTCOM            ;LOAD H/L WITH CONTROL COMMAND
        CMA                      ;INVERT DATA FOR BUSS
        MOV M,A                  ;ACTIVATE RESET
        MVI A,INITIAL            ;SET UP INITIALIZE CIMMAND
        CMA                      ;INVERT DATA FOR BUSS
        MOV M,A                  ;ACTIVATE INITIALIZE
        LXI H,IFLAG              ;LOAD H/L WITH IFLAG ADDRESS
        MVI A,01                 ;
        MOV M,A                  ;SET IFLAG = 1
        LXI H,BYTECNT            ;LOAD H/L WITH ADDRESS OF BYTE COUNT
        MVI A,00                 ;
        MOV M,A                  ;ZERO BYTE COUNT
        LXI H,PAGENUM            ;LOAD H/L WITH ADDRESS OF PAGE NUMBER
        MOV M,A                  ;ZERO LO BYTE, PAGE NUMBER
        INX H                    ;
        MOV M,A                  ;ZERO HI BYTE
        RET                      ;

BUSYCHK:MVI C,'3'                ;
        CALL CO                  ;

        MVI C,BUSY               ;SET UP BUSY CHECK BYTE
        LXI H,RDSTATUS           ;SET UP STATUS READ ADDRESS
        MOV A,M                  ;READ STATUS
        CMA                      ;COMPLIMENT ACCUMULATOR DUE TO REVERSE
                                 ;DATA LINE LOGIC LEVEL
        LXI H,STATSV             ;SET UP STATUS SAVE
        MOV M,A                  ;STORE CURRENT STATUS
*
```

```
ANI 10H                  ;MASK ALL BUT BUSY BIT
CMP C                    ;CHECK FOR BUSY
JZ BUSYCHK               ;LOOP IF BUSY
RET                      ;ELSE RETURN

ORG VRBLS                ;VARIABLE STORAGE AREA

WRBUFF  DB      0        ;WRITE BUFFER
IFLAG   DB      0        ;INITIALIZATION FLAG
BYTECNT DB      0        ;BYTES CURRENTLY READ/WRITTEN
PAGENUM DW      0        ;NUMBER OF PAGES CURRENTLY READ OR
                         ;WRITTEN.
STATSV  DB      0        ;CURRENT CONTROLLER STATUS


END 0C00H                ;
```

```
;        BUBBLE TEST PROGRAM-INTERRUPT CONTROLLED



ORG 3000H                    ;THIS PROGRAM,THROUGH INTERRUPT DRIVEN
                    ;ROUTINES, UTILIZES THE POWER OF THE BUBBLE CON-
                    ;TROLLER TO OVERCOME BUBBLE ACCESS TIME.  CON-
                    ;TROL PASSES TO THE OUTER PROGRAM LEVEL DURING
                    ;BUBBLE ACCESSES TO RETURN THIS 'LOST' TIME
                    ;TO THE MAIN LINE ROUTINE.  PROGRAM IS ORGED
                    ;AT 3000H(SBC RAM) TO ALLOW EASY MODIFICATION.


VRBLS EQU 3F00H              ;VARIABLE STORAGE AREA

LXI SP,3F80H                 ;STACK FROM 3F55H THRU 3F80 INCLUSIVE

TRUE EQU 0FFH                ;
FALSE EQU 0                  ;
LDPGRGLO EQU 0FFF0H          ;LOAD PAGE SELECT REGISTER, LO BYTE
LDPGRGHI EQU 0FFF1H          ;LOAD PAGE SELECT REGISTER, HI BYTE
CONTCOM EQU 0FFF2H           ;READ CONTROL COMMANDS FROM DATA BUSS
RDBYTE EQU 0FFF3H            ;READ DATA BYTE FROM BUBBLE
WRBYTE EQU 0FFF4H            ;WRITE DATA BYTE TO BUBBLE
RDSTATUS EQU 0FFF5H          ;READ STATUS REGISTER
PGCNTLO EQU 0FFF6H           ;READ/WRITE NUMBER OF S FOR MULTI-PAGE
                             ;PAGE TRANSFERS, LO BYTE
PGCNTHI EQU 0FFF7H           ;SAME FOR HI BYTE
LDMINSZLO EQU 0FFF8H         ;LOAD LOW BYTE, MINOR LOOP SIZE
                             ;(INITIALIZATION ONLY)
LDMINSZHI EQU 0FFF9H         ;SAME FOR HI BYTE
PGPOSLO EQU 0FFFAH           ;PAGE POSITION COUNTER, LOW BYTE. USED
                             ;TO SHOW CURRENT PAGE AT TRANSFER GATE
                             ;IN THE BUBELE
PGPOSHI EQU 0FFFBH           ;SAME FOR HI BYTE
LDPGSZRG EQU 0FFFCH          ;LOAD PAGE SIZE REGISTER
RDROMBYTE EQU 0FFFDH         ;READ CURRENT REDUNDENCY MAP,
                             ;INCREMENT POINTER
RWSTADDR EQU 0FFFEH          ;READ REDUNDENCY MAP ADDRESS

INITIAL EQU 1                ;INITIALIZE THE CONTROLLER
RDPG EQU 2                   ;READ PAGE,SINGLE PAGE MODE
WRPG EQU 4                   ;WRITE PAGE, SINGLE PAGE MODE
SGLPGMD EQU 8                ;SET SINGLE PAGE MODE
MULPGMD EQU 10H              ;SET MULTI-PAGE MODE
TESTMD EQU 20H               ;SET TEST MODE
RESET EQU 40H                ;SOFTWARE RESET
```

```
        INTMSK EQU 80H              ;SET CONTROLLER INTERRUPT MASK

        MONITOR EQU 08H             ;SET MONITOR CALL ADDRESS
        CO EQU 00FH                 ;CONSOLE OUT CALL ADDRESS
        CI EQU 012H                 ;CONSOLE IN CALL ADDRESS
        RI EQU 015H                 ;READER IN CALL ADDRESS
        PO EQU 018H                 ;PUNCH OUTPUT CALL ADDRESS
        ICCP EQU 0DAH               ;INTERRUPT COMMAND PORT
        MSKPT EQU 0DBH              ;INTERRUPT MASK PORT
        ICW1 EQU 16H                ;DEFINE INTERRUPT INITIALIZATION
        IMASK EQU 0                 ;INTERRUPT MASK
        EOIC EQU 20H                ;END-OF-INTERRUPT COMMAND WORD

        BUSY EQU 10H                ;DEFINE BUSY CHECK BYTE


        MINLPSIZ EQU 641D           ;NUMBER OF POSITIONS ON MINOR LOOP
        PAGESIZE EQU 18D            ;PAGESIZE IN BYTES

        LXI H,IFLAG                 ;FETCH IFLAG
        MVI A,0                     ;ZERO ACCUMULATOR
        MOV M,A                     ;ZERO FLAG1


        CALL CI                     ;
        MOV C,A                     ;SET UP ECHO
        CALL CO                     ;ECHO OUT
        MVI B,0D2H                  ;CONSOLE INPUT FOR CAP R
        CMP B                       ;SEE IF R INPUT IMPLYING READ
        JZ PAGERD                   ;IF SO, GO TO READ ROUTINE. ELSE ENTER
                          ;SIMULATED 'OUTER LEVEL' FOR ACCESS TO INT-
                          ;ERRUPT DRIVEN WRITE ROUTINE.


LOOP:   LXI H,BUBBLE                ;SET UP LOOP TO WAIT FOR INT 6 BEFORE
                                    ;ALLOWING INPUT
        SHLD 3FF9H                  ;MODIFY INTERRUPT 6 VECTOR IN RAM
        LXI H,RINPT                 ;SET UP INTERRUPT 5 VECTOR
        SHLD 3FF5H                  ;WRITE INTO RAM
        MVI C,'L'                   ;
        CALL CO                     ;OUTPUT LOOP TRACE
        NOP
        NOP
        NOP

        JMP LOOP                    ;LOOP FOREVER. INT 6 WILL TAKE US TO
                                    ;BUBBLE. PGREAD ACCESSED VIA KEYBOARD

BUBBLE: CALL SAVE                   ;SAVE MAIN PROGRAM STATUS
        NOP
*
```

```
NOP
NOP

CALL INIT                  ;INITIALIZE IF RQD


LHLD PAGENUM               ;LOAD UP DESIRED PAGE
XCHG                       ;
LXI H,LDPGRCLD             ;LOAD UP COMMND
MOV A,E                    ;
CMA                        ;INVERT FOR DATA BUSS
MOV M,A                    ;OUTPUT LO BYTE OF PAGE
INX H                      ;INCREMENT COMMAND
MOV A,D                    ;
CMA                        ;SAME
MOV M,A                    ;OUTPUT HI BYTE COMMAND

MVI C,'*'                  ;
CALL CO                    ;OUTPUT TRACE
CALL CI                    ;INPUT DATA FROM CONSOLE
LXI H,WRBUFF               ;SET UP STORE
MOV M,A                    ;STORE INPUT DATA IN BUFFER
MOV C,A                    ;SET UP ECHO
CALL CO                    ;ECHO INPUT


CALL BUSYCHK               ;SEE IF CONTROLLER DONE
LXI D, WRBUFF              ;
LDAX D                     ;MOVE INPUT DATA BACK INTO ACCUMULATOR

LXI H,WRBYTE               ;SET UP WRITE COMMAND
CMA                        ;INVERT FOR DATA BUSS
MOV M,A                    ;WRITE DATA BYTE TO FIFO
LXI H,BYTECNT              ;SET UP BYTE COUNT INCREMENT
INR M                      ;INCREMENT BYTE COUNT(INDIRECT)
MVI A,PAGESIZE             ;
CMP M                      ;
CZ PGWRT                   ;IF BYTES WRITTEN = PAGE
                           ;SIZE, WRITE PAGE
MVI C,'8'                  ;
CALL CO                    ;TRACE FOR RETURN FROM PAGE WRITE
NOP
NOP
NOP
DI                         ;DISABLE INTERRUPTS AROUND 8259
                           ;COMMANDS
MVI A,EOIC                 ;
OUT ICCP                   ;OUTPUT END OF INTERRUPT 6
EI                         ;RE-ENABLE THEM
CALL RESTORE               ;RESTORE MAIN PROGRAM STATUS
```

```
        RET                         ;RETURN FROM INTERRUPT

PGWRT:  MVI C, '4'                  ;
        CALL CO                     ;OUTPUT TRACE
        NOP
        NOP
        NOP

        CALL BUSYCHK                ;
        LXI H,CONTCOM               ;
        MVI A,INTMSK                ;UNMASK CONTROLLER INTERRUPT TO ALLOW-
        CMA                         ;INVERT FOR DATA BUSS
        MOV M,A                     ;-INTERRUPTS TO BE OUTPUT TO CPU
        MVI A,WRPG                  ;
        CMA                         ;INVERT FOR DATA BUSS
        MOV M,A                     ;PAGE WRITE COMMAND TO CONTROLLER
        RET                         ;RETURN TO OUTER LEVEL PROGRAM WHILE
                         ;CONTROLLER WORKS.


RINPT:  CALL SAVE                   ;SAVE MAIN PROGRAM STATUS ON SECOND
                         ;ENTRY, WHILE COMPLETING BUBBLE CYCLE.
ENDCK:  MVI C,'5'                   ;SET UP & OUTPUT TRACE
        CALL CO                     ;
        NOP
        NOP
        NOP
        CALL BUSYCHK                ;ENSURE CONTROLLER DONE
        MVI A,0                     ;ZERO ACCUMULATOR
        LXI H,CONTCOM               ;
        CMA                         ;INVERT FOR DATA BUSS
        MOV M,A                     ;RESET INTERRUPT MASK
        LHLD PAGENUM                ;LOAD ADDRESS OF PAGE NUMBER
        INX H                       ;INCREMENT PAGE NUMBER
        SHLD PAGENUM                ;RESTORE INCREMENTED PAGE NUMBER
        LXI H,PAGENUM               ;
        MOV E,M                     ;PUT HI BYTE OF PAGE NUMBER IN REG E
        INR L                       ;INCREMENT ADDRESS
        MOV D,M                     ;LO BYTE IN D
        LXI H,MINLPSIZ              ;
        MOV A,H                     ;HI BYTE INTO ACCUM
        CMP D                       ;CHECK FOR HI BYTE OF PAGE NUMBER = HI
                                    ;BYTE OF MINOR LOOP SIZE
        CZ LOCHK                    ;IF HI BYTES EQUAL, CHECK LO BYTES
        LXI H,BYTECNT               ;LOAD BYTE COUNT STORAGE LOCATION INTO
                                    ;THE H/L
        MVI A,0                     ;
        MOV M,A                     ;RESET THE BYTE COUNT TO ZERO
        LXI D,PAGENUM               ;SET UP TO LOAD PAGE SELECT REGISTER
```

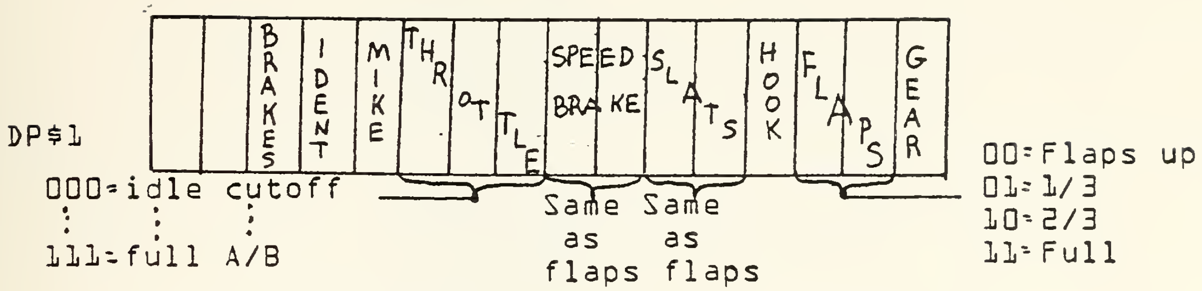
```
        LDAX D                  ;LOAD ACCUM FROM ADDRESS IN D/E
        LXI H,LDPGRGLO          ;LOAD H/L WITH LOAD PAGE REGISTER LO
                                ;COMMAND ADDRESS
        CMA                     ;INVERT FOR DATA BUSS
        MOV M,A                 ;OUTPUT PAGE TO PAGE SELECT REGISTER
                                ;(LO BYTE)
        INX H                   ;
        INX D                   ;INCREMENT FOR HI BYTE
        LDAX D                  ;LOAD IN HI BYTE
        CMA                     ;INVERT FOR DATA BUSS
        MOV M,A                 ;OUTPUT PAGE TO PAGE SELECT REGISTER
                                ;(HI BYTE)
        CALL RESTORE            ;RESTORE MAIN PROGRAM STATUS
        RET                     ;
LOCHK:  MVI C, '9'              ;
        CALL CO                 ;OUTPUT TRACE
        NOP
        NOP
        NOP

        MOV A,L                 ;
        CMP E                   ;CHECK LO BYTES
        CZ ZEROPG               ;RESET PAGE TO ZERO
                                ;IF AT END OF BUBBLE(LOOP MEMORY)
        RET                     ;
ZEROPG: MVI C, 'A'              ;
        CALL CO                 ;OUTPUT TRACE
        NOP
        NOP
        NOP

        LXI H,PAGENUM           ;LOAD H/L WITH ADDRESS OF PAGE NUMBER
        MVI A,00                ;ZERO ACCUMULATOR
        MOV M,A                 ;ZERO HI BYTE, PAGE NUMBER
        INX H                   ;INCREMENT
        MOV M,A                 ;ZERO LO BYTE

        RET                     ;RETURN TO LOOP MEMORY

INIT:   MVI C,'2'               ;
        CALL CO                 ;OUTPUT TRACE
        NOP
        NOP
        NOP

        LDA IFLAG               ;SEE IF NEED TO INITIALIZE
        CPI 1                   ;SEE IF SET
        RZ                      ;IF INITIALIZED ALREADY, RETURN
```

```
        CALL BUSYCHK            ;
        MVI A,PAGESIZE          ;
        LXI H,LDPGSZRG          ;LOAD H/L WITH LOAD PAGE SIZE REGIS-
                                ;TER COMMAND
        CMA                     ;INVERT FOR DATA BUSS
        MOV M,A                 ;LOAD PAGE SIZE INTO PAGE SIZE REG
        LXI B,MINLPSIZ          ;
        MOV A,C                 ;MOVE MINOR LOOP SIZE LO BYTE TO ACCUM
        LXI H,LDMINSZLO         ;LOAD H/L WITH LOAD MINOR LOOP SIZE LO
                                ;COMMAND ADDRESS
        CMA                     ;INVERT FOR DATA BUSS
        MOV M,A                 ;LOAD MINOR LOOP SIZE LO BYTE INTO
                                ;MINOR LOOP SIZE REGISTER
        MOV A,B                 ;SET UP HI BYTE
        INX H                   ;
        CMA                     ;INVERT FOR DATA BUSS
        MOV M,A                 ;LOAD MINOR LOOP SIZE, HI BYTE
        MVI A,RESET             ;SET UP RESET COMMAND
        LXI H,CONTCOM           ;LOAD H/L WITH CONTROL COMMAND
        CMA                     ;INVERT FOR DATA BUSS
        MOV M,A                 ;ACTIVATE RESET
        MVI A,INITIAL           ;SET UP INITIALIZE CIMMAND
        CMA                     ;INVERT FOR DATA BUSS
        MOV M,A                 ;ACTIVATE INITIALIZE
        LXI H,IFLAG             ;LOAD H/L WITH IFLAG ADDRESS
        MVI A,01                ;
        MOV M,A                 ;SET IFLAG = 1
        LXI H,BYTECNT           ;LOAD H/L WITH ADDRESS OF BYTE COUNT
        MVI A,00                ;
        MOV M,A                 ;ZERO BYTE COUNT
        LXI H,PAGENUM           ;LOAD H/L WITH ADDRESS OF PAGE NUMBER
        MOV M,A                 ;ZERO LO BYTE, PAGE NUMBER
        INX H                   ;
        MOV M,A                 ;ZERO HI BYTE
        RET                     ;

SAVE:   PUSH H                  ;SAVE H/L
        PUSH D                  ;SAVE D/E
        PUSH B                  ;SAVE B/C
        PUSH PSW                ;SAVE ACCUMULATOR AND STATUS FLAGS
        RET

RESTORE:POP PSW                 ;RESTORE ACCUMULATOR AND STATUS FLAGS
        POP B                   ;RESTORE B/C
        POP D                   ;RESTORE D/E
        POP H                   ;RESTORE H/L
        RET                     ;

BUSYCHK:MVI C,'3'               ;
        CALL CO                 ;
*
```

```
        MVI C,BUSY               ;SET UP BUSY CHECK BYTE
        LXI H,RDSTATUS           ;SET UP STATUS READ ADDRESS
        MOV A,M                  ;READ STATUS
        CMA                      ;COMPLIMENT DATA TO ACCOUNT FOR
                         ;REVERSE LOGIC DATA LINES
        ANI 10H                  ;MASK ALL BUT BUSY BIT
        CMP C                    ;CHECK FOR BUSY
        JZ BUSYCHK               ;LOOP IF BUSY
        RET                      ;ELSE RETURN


PAGERD: MVI C,'&'                ;
        CALL CO                  ;OUTPUT TRACE
        CALL INIT                ;INITIALIZE IF RQD
PGRD:   CALL BUSYCHK             ;
        LXI H,CONTCOM            ;
        MVI A,RDPG               ;
        CMA                      ;INVERT FOR DATA BUSS
        MOV M,A                  ;OUTPUT READ PAGE COMMAND. A PAGE OF
                                 ;BUBBLE DATA WILL NOW BE READ INTO THE
                                 ;FIFO UNDER CONTROL OF THE CONTROLLER
        CALL BUSYCHK             ;WAIT TILL DONE

BYTERD: LXI H,RDBYTE             ;
        MOV A,M                  ;TRANSFER IN A BYTE FROM THE FIFO
        CMA                      ;INVERT INCOMING DATA
        ADI 30H                  ;ASCII ENCODE FOR DISPLAY
        MOV C,A                  ;TRANSFER TO C FOR OUTPUT
        CALL CO                  ;OUTPUT CHARACTER

        LXI H,BYTECNT            ;
        INR M                    ;INCEMENT BYTE COUNTER
        MVI A,PAGESIZE           ;
        CMP M                    ;CHECK TO SEE IF ALL OF FIFO TRANS-
                                 ;FERRED
        JNZ BYTERD               ;IF NOT DONE, LOOP
        CALL ENDCK               ;IF DONE, INCREMENT AND ZERO
        JMP PGRD                 ;LOOP UNTIL ENDCK CAUSES HALT

        ORG VRBLS                ;VARIABLE STORAGE AREA

        WRBUFF  DB       0       ;WRITE BUFFER
        IFLAG   DB       0       ;INITIALIZATION FLAG
        BYTECNT DB       0       ;BYTES CURRENTLY READ/WRITTEN
        PAGENUM DW       0       ;NUMBER OF PAGES CURRENTLY READ OR
                                 ;WRITTEN.

        END 3000H                ;
*
```

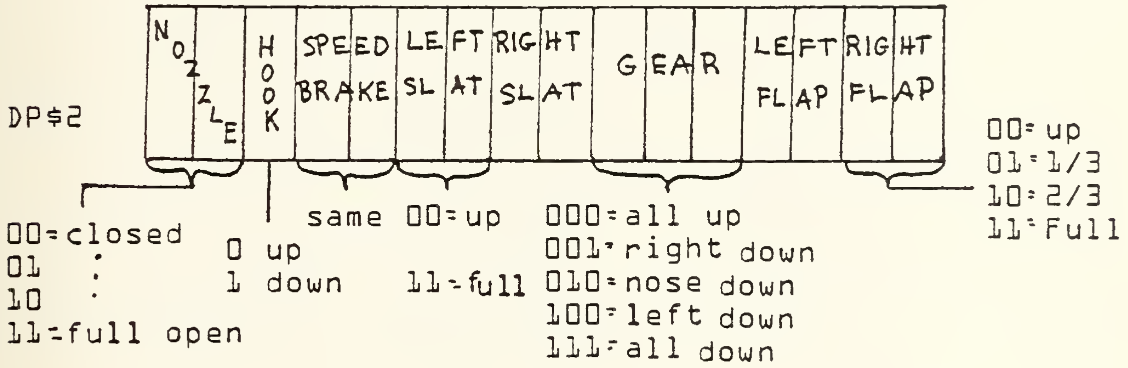
TABLE IV

DISCRETE PARAMETERS



16 BIT DISCRETE PARAMETER 1
Pilot inputs

EXAMPLE: DP$1 = 0000110000111111 = 0C3Fh is a current pilot input of geardown, full flaps and slates, hook down, no speed, brake, intermediate throttle, with the pilots mike keyed.



EXAMPLE: DP$2 = 0001111010000011 = 1E83h is a current aircraft state in which the engine nozzle is full closed, hook is up, speed brake is fully extended, the left slat is full down, the right slat is 1/3 down, all gear indicate up, the left flap is full up, and the right flap is full down.

INITIALIZATION

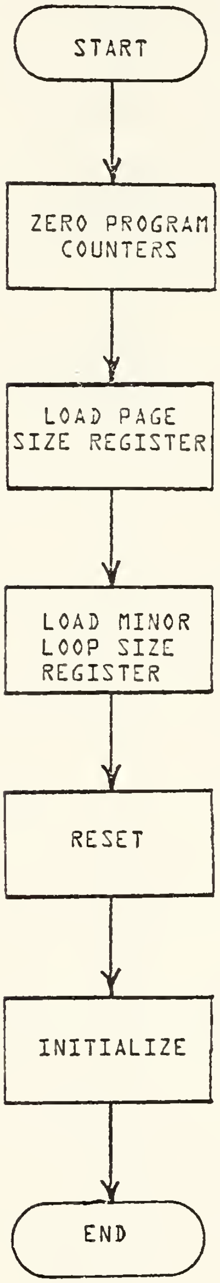


FIGURE 15

SINGLE PAGE READ OR WRITE

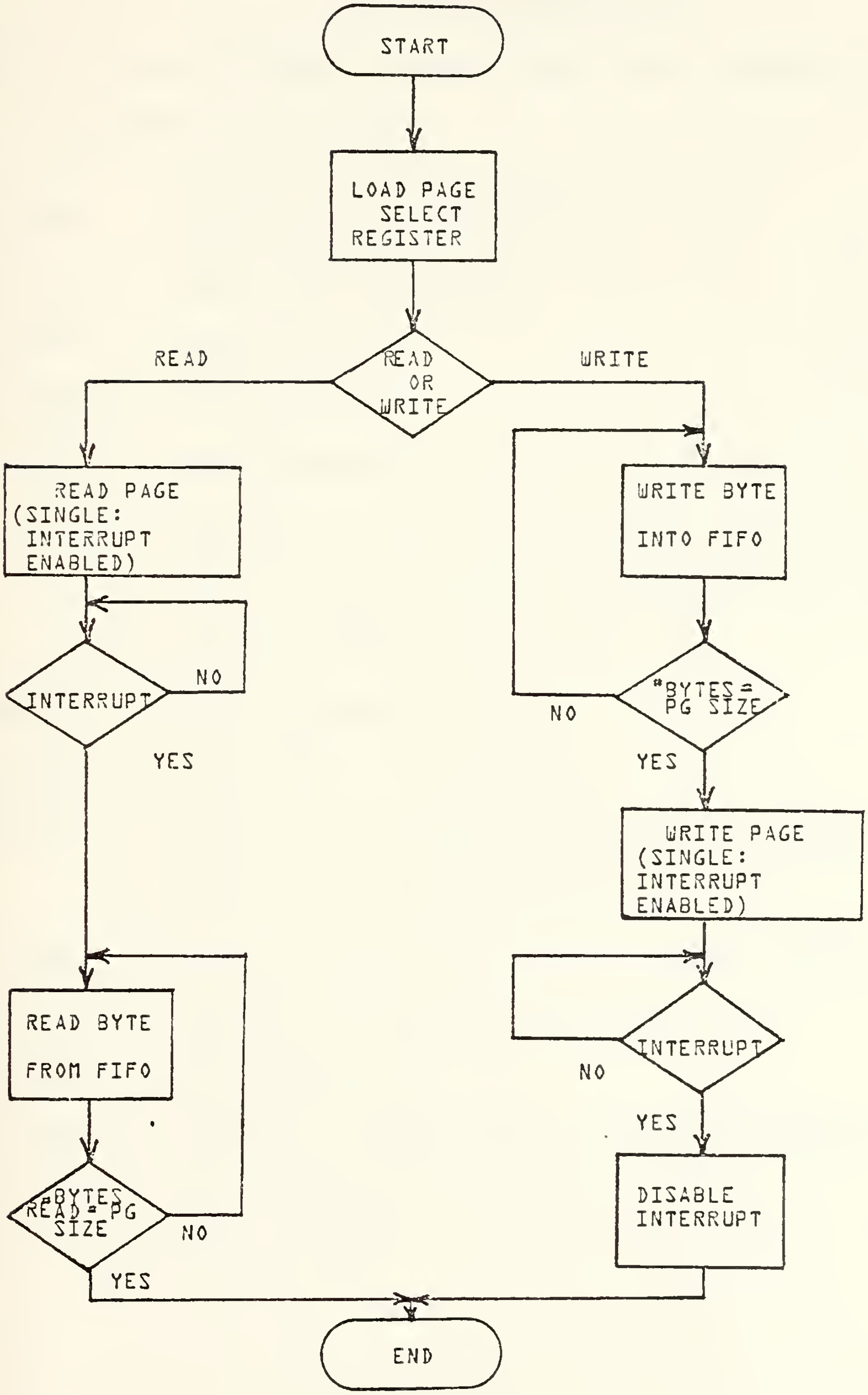


FIGURE 16

TABLE V

DATA RECORDER MEMORY MAP

NOTE: Includes ICOM PROM programmer board with resident monitor, and controller memory usage.

0000-06AE	EPROM system 80/20-4 monitor	(chips 0,1)
06AF-07FF	Remaining usable EPROM	(chip 1)
0800-0BFF	Usable EPROM	(chip 2)
0C00-0FFF	Usable EPROM	(chip 3)
1000-1FFF	Not implemented; may be utilized to double low EPROM memory by switching to Intel 2716 (2KX8) EPROMS in the first four sockets with appropriate adjustments (Reference [3], pp. 2-15).	
2000-2FFF	Not implemented	
3000-3CTF	User RAM	
3C20-3015	User RAM: also used by ICOM PROM programmer monitor when control passed to programmer.	
3D16-3F80	User RAM. Stack must start at 3F80 (push down).	
3F81-3FFF	SBC 80/20-4 monitor reserved RAM.	
4000-BFFF	Not implemented	
C000-DBFF	7K BYTES of EPROM (2708) memory on ICOM PROM programmer board, available for program use.	

DC00-DFFF	Programmer Monitor. If this last 1K is needed, the programmer monitor may be removed and another EPROM put in its place.
C000-EFFF	Not implemented.
FFF0-FFFE	Bubble controller mapped in this area.
FFFF	Not implemented.

TABLE VI

MDS MEMORY MAP UNDER ICE/80

0000-0023	ISIS interrupts (0-2)	ICE80 interrupts
0024-0031	ICE80 interrupt (3)	
0032-0063	User interrupts (4-7)	
0064-2FFF	ISIS resident area	
3000-5FFF	Ice 80 resident area	
6000-F6BF	User RAM. Note: User symbol table in the top of user RAM	
FGC0-F7FF	318 locations for ICE 80 variables	
F800-FFFF	MDS monitor (64K contiguous RAM)	

Based on this map, the only MDS memory available to the ICE 80 X Form Memory Commands are blocks 6-E inclusive (36K).

APPENDIX C

CONSTRUCTION NOTES

A. GENERAL

Pertinent construction details not found in listed references follow. Wiring diagrams for all cabling and special circuitry are included. Component placement on the prototype board is indicated. Specific hardware related problems unsolved at this time are reviewed.

B. MICROCOMPUTER

The system 80/20 was utilized essentially as shipped. The following changes were incorporated to facilitate MBM module interface.

1. Fail Safe Timer

The Fail Safe Timer input to the Ready Circuit was disabled by removal of the jumper between taps 137-138. The function of the Fail Safe Timer is to supply, in default, the acknowledgement signal when an off board memory access is made, to prevent CPU hang-ups due to software error. The timer will furnish this signal, if no other device does, after 10ms of wait time. The controller utilizes its ready output to control this circuit. The ready signal is output high to the interface board, to a 4.7K pull up resistor tied to ±5V. This is then inverted and tied to the Transfer In (SBC P1/23) line to the master bus controller, furnishing a continuous acknowledge signal. If the controller requires additional time, it pulls its ready line low, which in turn pulls the CPU ready line low. This is repeated as often as required.

Removal of jumper 137-138 is not required for system operation after the test phase, and should be re-installed to return full capability to the SBC.

2. EIA Interface

Pins eight and ten of SBC plug J 3 were jumper shorted on the SBC card back to provide EIA Clear To Send to the Terminal output (Reference [3], pp. 2-20).

3. Interrupt Implementation

Taps 31 and 36-39 were jumped together to allow use of interrupts, without utilizing all of interrupt seven's inputs. This is described in Reference [1], pages 2-16. Additionally, interrupts five and six were implemented by jumping taps 21 to 48, and 30 to 49 respectively.

4. Master Bus

Once the system 80/20 mother board, J5/15 was jumpered to J4/15 to provide an interface board ground to "lock" the SBC onto the bus as bus master whenever the interface board is inserted. This supplies a ground to the SBC Bus Priority In (BPRN/) signal and saves bus access time every time the controller is accessed. This same gain may be realized by outputting the Bus Overide signal to the Master Bus Controller via Software (Reference [10], pp. 4-7: "Multibus Overide").

5. System Reset

Reset was jumped from J5A/38 to J4A/38 to provide System Reset to the interface board and magnetic bubble module.

C. CONTROLLER BOARD

Controller board implementation is as described in Chapter II and Reference 7, except as indicated below.

1. Interrupt

The low level interrupt from the controller is not carried across the board to the master bus. This should be wired across to interrupt 5/.

2. Counters

U9 and U5 are SN74LS163A and need to be replaced with SN74S163A. The lower power Schottky version of these counters was not sufficiently fast and must be replaced with the straight Schottky version. TI indicated it will supply these in the near future.

3. ROMS

U1 and U2 may need to be replaced. Validation of ROM contents to ensure they agree with Section II must be accomplished. Validation of U11, the redundancy mask might also be checked.

4. Addressing

Address Lines listed on Reference 7 are implemented differently. Table VII indicates the correspondence.

5. Wiring

Jumper Pad 14 is connected to Pad 15. This is omitted on the schematic (Reference [7]).

6. Power Failure

The power failure circuitry is not connected between the system 80/20 and the controller input.

D. MBM BOARD

Magnetic Bubble Memory Board implementation is as listed in Reference [8] with the following exceptions:

1. Capacitors

C1-C12 are incorrect. This was discovered quite late in the research. The incorrect milspec part was supplied. The current capacitors are much too small. The correct part number is M39014/01-1473. These parts have been ordered.

2. Diodes

The Schottky diodes were unavailable in the part number indicated. A higher voltage rating unit was substituted.

E. INPUT/OUTPUT

A Texas Instruments Silent 700 model 745 was obtained for a field test terminal. As shipped it is not configured for this purpose and was modified to interface to the SBC. Internal jumpers were removed to disable the acoustic coupler inputs for TX Data, RX Data, and Data Carrier Detect. This is required to prevent external noise input during operation. An interface cable was then constructed to operate the unit.

This cable is equally compatible with the Intel MDS 800 to allow for maximum peripheral interchangability during development. Operation of the MDS with the Silent 700 is implemented at 300 BAUD. The Monitor Module has been modified to allow switch selection of 300 or 2400 Baud rate for the CRT interface, as the silent 700 printer is an EIA CRT interface. For 300 BAUD, switch one is on, with the other four off. Similarly for 2400 BAUD, switch five must be on alone.

Operation of the SBC with the Silent 700 is slightly different in that the keyboard is not implemented for lower case ASCII letters. As such, for automatic BAUD rate selection after SBC reset, the shift key

is not utilized to obtain an upper case "V". Simply depress the "V" key six times to receive the monitor sign on message.

Operation of the terminal thrugh the usual acoustic coupler is still possible, as a jumper plug was fabricated. Installed in J1, in place of the interface cable, it restores the Silent 700 to original configuration. The jumper is constructed by shorting P1/11 to P1/2, P1/12 to P1/8, and P1/13 to P1/3.

F. EXTERNAL SUPPLY

Reference [6] indicates the MBM board will operate on ±5 VDC and ±12 VDC. This was a major factor in the decision to undertake MBM research in connection with the construction of the data recorder, as the SBC power bus provides only these voltages. In fact, an additional voltage of +17 VDC is required. TI indicates this will be designed out in future board designs. For now this requires an external supply.

Analysis of the SBC power supply indicates it might be possible to tap the unregulated output of the +18 VDC winding and construct a regulated +17 VDC supply within the System 80/20 frame. Schematics were obtained but this has not yet been carried out.

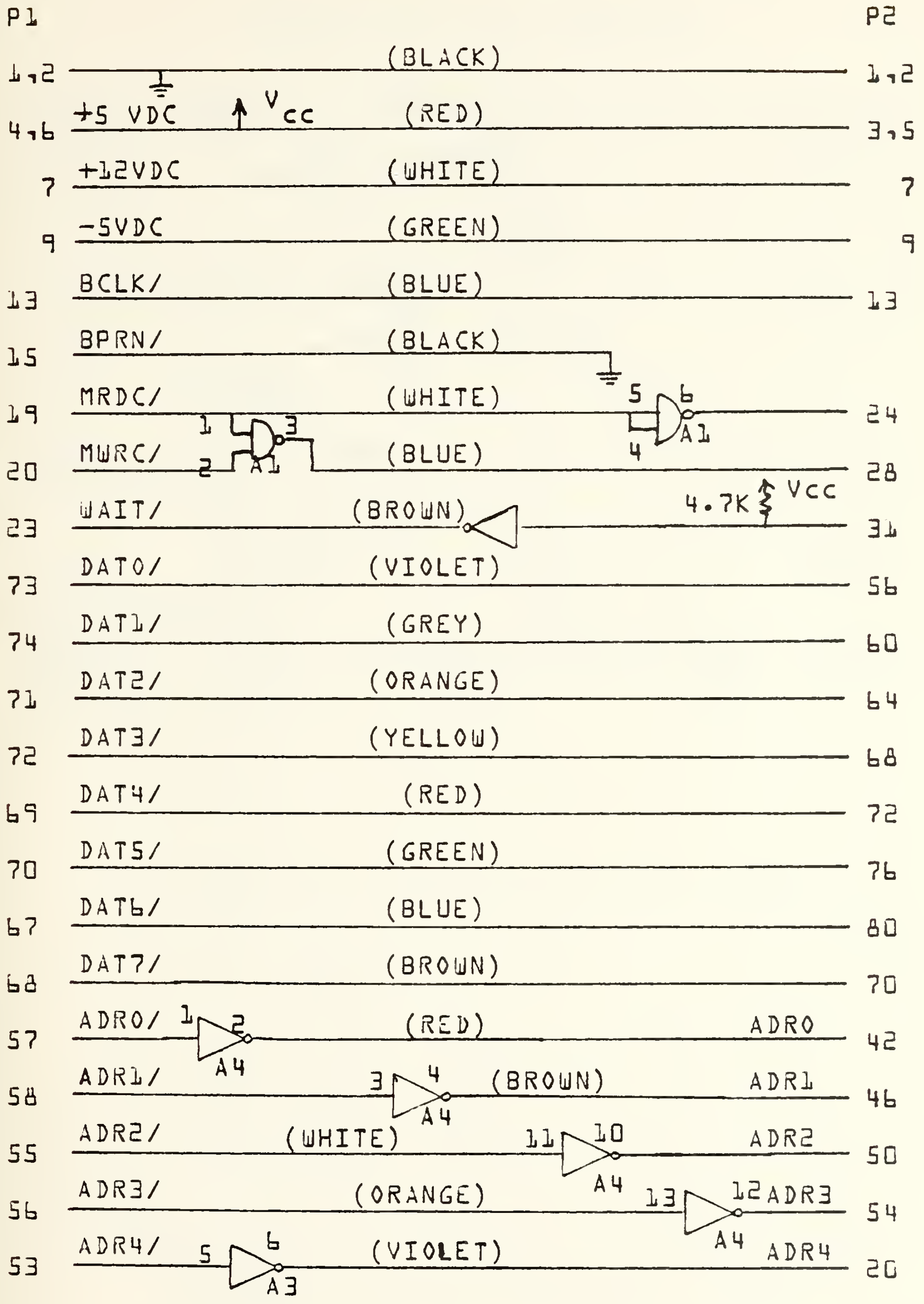
TABLE VII

ADDRESS LINE CORRESPONDENCE

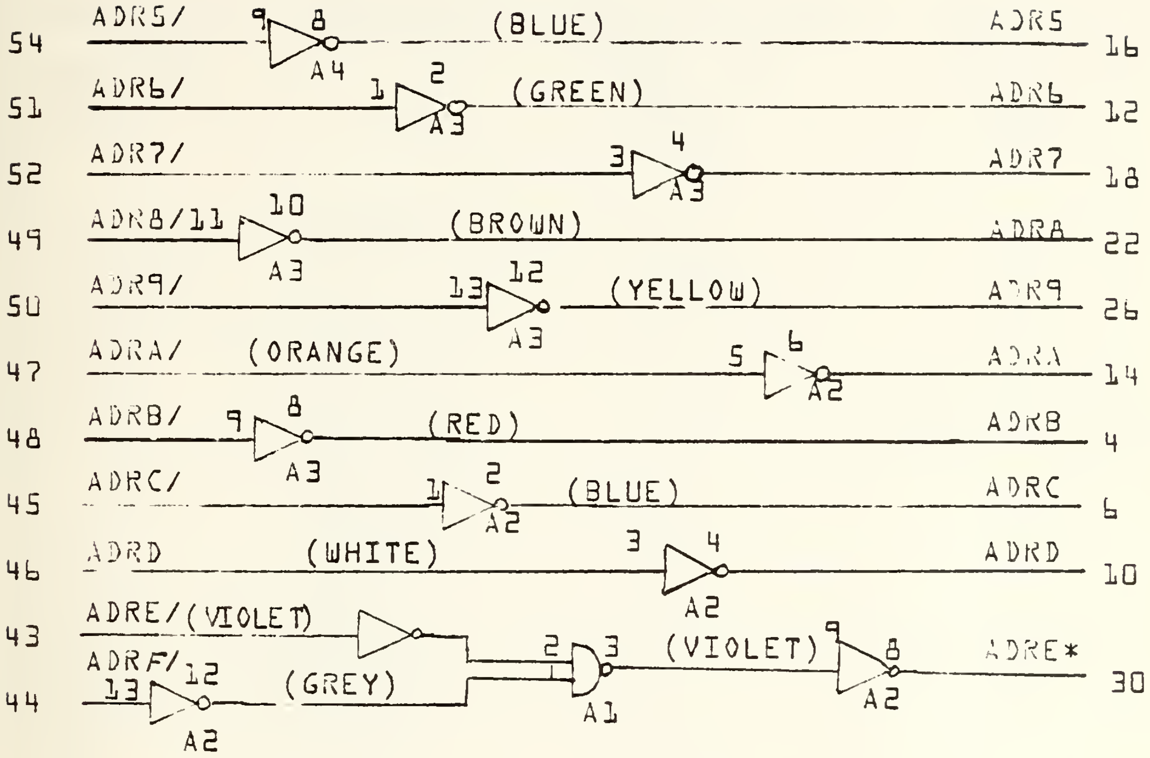
SCHEMATIC ADDRESS LINE	PIN NO. P2	MASTER BUS ADDRESS
14	21	0
13	23	1
12	25	2
11	27	3
2	10	4
1	8	5
0	6	6
10	9	7
9	11	8
8	13	9
7	7	A
6	2	B
5	3	C
4	5	D
3	15	E and F

TABLE VIII

INTERFACE BOARD SCHEMATIC



INTERFACE BOARD SCHEMATIC



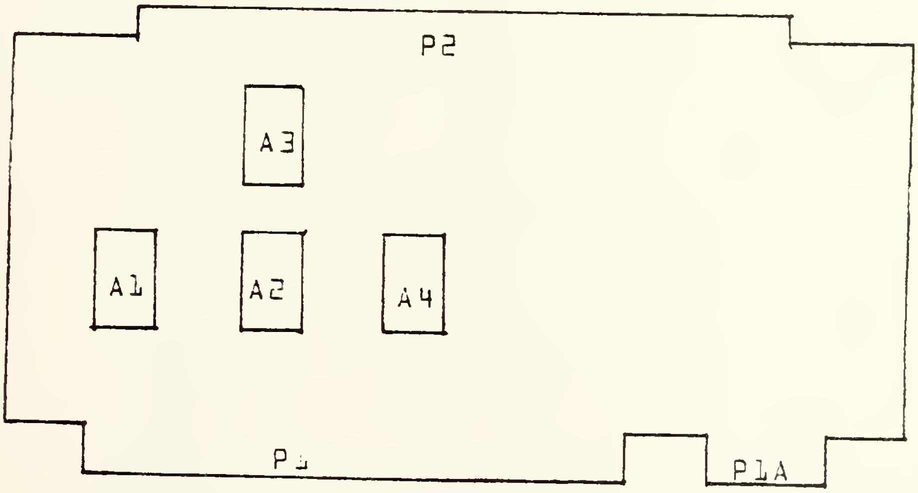


TABLE IX

CONTROLLER/BUBBLE CAGE BACKPLANE

MBM BOARD PLUG P1 PIN*	FUNCTION	CONTROLLER BOARD PLUG P1 PIN*	INTERFACE BOARD PLUG P2 PIN*
1,A	GROUND	1	1
2,21,B,Y	+5 V	21	5
3,11,12,13	+12 VDC	3	4
C,M,N,	+12 VDC	3	4
4,D,P	+17 VDC	D	EXTERNAL SUPPLY
5	DAT OUT	5	
6	XOUT/	6	
7-8	UNUSED	---	---
9	ANN/	9	
10	UNUSED	---	---
14-15	UNUSED	---	---
16	CYA/	16	
17	XIN/	14	
18	CXB/	17	
19	CLAMP/	11	
20	STROBE/	13	
22,Z	-5 VDC	22	2&3
E	BDEN/	1	
F-J	UNUSED	---	---
K	REP/	K	
L	UNUSED	---	---
R	UNUSED	---	---
S	CYB/	S	
T	GEN/	12	
U	CXA/	T	
V-X	UNUSED	---	---

TABLE X

CABLE-INTERFACE BOARD TO CONTROLLER

INTERFACE BOARD
PLUG P-2/CABLE
SOCKET J2-A

CONTROLLER PLUG
P2 CABLE SOCKET
J2-B

PIN *	FUNCTION	COLOR	PIN*
A-1	SIGNAL GROUND	BROWN	1
A-2	ADDRESS (B)	RED	2
A-3	ADDRESS (C)	ORANGE	3
A-4	BOARD SELECT(A)	YELLOW	4
A-5	ADDRESS (D)	GREEN	5
A-6	ADDRESS(6)	BLUE	6
A-7	ADDRESS(A)	VIOLET	7
A-8	ADDRESS(5)	GREY	8
A-9	ADDRESS(7)	WHITE	9
A-10	ADDRESS(4)	BLACK	10
A-11	ADDRESS(8)	BROWN	11
A-12	DBIN	RED	12
A-13	ADDRESS(9)	ORANGE	13
A-14	MEMEN	YELLOW	14
A-13	ADDRESS(E)	GREEN	15
A-16	BOARD SELECT(C)	BLUE	16
A-17	HARDWARE RESET	VIOLET	17
A-18	BOARD SELECT(B)	GREY	18
A-19	POWER BAD	WHITE	19
A-20	----------	BLACK	20
A-21	ADDRESS(0)	BROWN	21
A-22	----------	RED	22
A-23	ADDRESS(1)	ORANGE	23
A-24	----------	YELLOW	24
A-25	ADDRESS(2)	GREEN	25

PIN*	FUNCTION	COLOR	PIN*
A-26	-----------	BLUE	26
A-27	ADDRESS(3)	VIOLET	27
A-28	DATA(0)	GREY	28
A-29	CLOCK (Φ2)	WHITE	29
A-30	DATA(1)	BLACK	30
A-31	READY TO MICROPROCESSOR	BROWN	31
A-32	DATA(2)	RED	32
A-33	-----------	ORANGE	33
A-34	DATA(3)	YELLOW	34
A-35	DATA(7)	GREEN	35
A-36	DATA(4)	BLUE	36
A-37	-----------	VIOLET	37
A-38	DATA(5)	GREY	38
A-39	INTERRUPT TO CPU	WHITE	39
A-40	DATA(6)	BLACK	40

CONNECTIONS FROM J2-A/B TO CARDCAGE BACK PLANE

			MBM BOARD PIN *
B-1	GROUND	BLACK	1
B-2	+5VDC	RED	22
B-3	+5VDC	RED	22
B-4	+12VDC	WHITE	12
B-5	-5VDC	GREEN	2
---	UNUSED	BLUE	---
EXTERNAL SUPPLY	+17V	BROWN(PURPLE)	4

NOTE: J2-A IS A 100 PIN EDGE CARD CONNECTION.
J2-B IS A 40 PIN A-D PRODUCTS CONNECTOR.

BIBLIOGRAPHY

1. Baetz, L. N., Study and Design of Flight Data Recording Systems for Military Aircraft, Master of Science Thesis, Naval Postgraduate School, June 1976.

2. Albertolli, W. R., Microcomputer Based Solid State Crash Data Recorder for Military Aircraft, Master of Science Thesis, Naval Postgraduate School, December 1976.

3. Intel, System 60/20-4 Microcomputer Hardware Reference Manual, Preliminary 1977.

4. ICOM Model PP80 MDS/SBC 80/20-R PROM Programmer Operatives Manual and Software Guide Listing, 1977.

5. Texas Instruments TMS 9916/5502 Controller Spec. Identification No. 96214, November 17, 1975, with changes through April 4, 1977.

6. Texas Instruments Magnetic Bubble Memories and System Interface Circuits, February 1977.

7. Texas Instruments Drawing No. SK326758, Logic Diagram, Bubble Memory Controller, May 10, 1977.

8. Texas Instruments Drawing No. BK 312763, Electronic Schematic Diagram, BK10103 Mag Bubble Memory Printed Circuit Evaluation Board, 1977.

   Texas Instruments Drawing No. SK312765, Assembly Drawing BK10103 Mag. Bubble Mem. Printed CXT Evaluation Board, 1977.

9. Texas Instruments, Model 745 Portable Data Terminal Operating Instructions Manual No. 984024-9701 Rev. A, Rev. 1 November 1976.

10. Intel, SBC 80/20 and SBC 80/20-4 Single Board Computer Hardware Reference Manual, 1977.

11. Air Force Avionics Laboratory Technical Report AFAL-TR-76-197, Magnetic Bubble Mass Memory, by R. A. Naden adn others, January 1977.

12. Air Force Avionics Laboratory Technical Report AFAL-TR-76-267, Small Bubble Techniques, by D. C. Bullock and others, May 1977.

13. Geusic, J. E., "Update on Bubble Memories," Electro, April 19, 1977.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
4.	Associate Professor U. R. Kodres, Code 52 (thesis advisor) Computer Science Department Naval Postgraduate School Monterey, California 93940	1
5.	LT Dennis L. Kane, USN SMC Box 1625 Naval Postgraduate School Monterey, California 93940	1
6.	Captain Clyde Tuomela, USN Aviation Safety Programs Naval Postgraduate School Monterey, California 93940	1

Thesis 173907
K136 Kane
c.1
Design and construction of a flight monitor and data recorder.

Thesis 173907
K136 Kane
c.1
Design and construction of a flight monitor and data recorder.

Design and construction of a flight moni
3 2768 002 11398 7
DUDLEY KNOX LIBRARY